

# Electronic computation of squared rectangles

### Citation for published version (APA):

Duijvestijn, A. J. W. (1962). Electronic computation of squared rectangles. [Phd Thesis 2 (Research NOT TU/e / Graduation TU/e), Mathematics and Computer Science]. Technische Hogeschool Eindhoven. https://doi.org/10.6100/IR44157

DOI: 10.6100/IR44157

### Document status and date:

Published: 01/01/1962

### Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

### Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.

• The final author version and the galley proof are versions of the publication after peer review.

• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- · Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
  You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

#### Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

### ELECTRONIC COMPUTATION OF SQUARED RECTANGLES

## ELECTRONIC COMPUTATION OF SQUARED RECTANGLES

### PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR IN DE TECHNISCHE WETENSCHAP AAN DE TECHNISCHE HOGESCHOOL TE EIND-HOVEN, OP GEZAG VAN DE RECTOR MAGNIFI-CUS Dr K. POSTHUMUS, HOOGLERAAR IN DE AFDELING DER SCHEIKUNDIGETECHNOLOGIE, VOOR EEN COMMISSIE UIT DE SENAAT TE VER-DEDIGEN OP VRIJDAG, 29 JUNI 1962, DES NA-MIDDAGS TE 4 UUR

DOOR

### ADRIANUS JOHANNES WILHELMUS DUIJVESTIJN

ELECTROTECHNISCH INGENIEUR GEBOREN TE 's-GRAVENHAGE

### DIT PROEFSCHRIFT IS GOEDGEKEURD DOOR DE PROMOTOR PROF. DR C. J. BOUWKAMP

•

#### Summary

This thesis considers problems that arise when the calculation of squared rectangles is automatized on an electronic computer. After the introductory chap. 1, it is indicated in chap. 2 how a so-called c-net is coded such that it can be processed by the computer. In particular, properties of the net in connection with its planarity are easy to recognize using this code. It is shown how, from the code of the original net, the code of the dual net can be obtained. Also described is how the branches and the vertexvertex incidence matrix of the net can be found from the code of the net. In chap. 3, a set of codes  $\Sigma_{N+1}$  representing c-nets of order N+1is generated from the set of codes  $T_N$  of all different c-nets of order N, by addition of a wire in the latter c-nets or their duals. The set  $\Sigma_{N+1}$ may contain codes representing one and the same net or its dual. Therefore a method is described how to each net a number can be assigned that characterizes the net uniquely. Sorting with respect to this characteristic number gives the set of codes  $T_{N+1}$  of all different c-nets of order N+1. Additional information as to whether the net is selfdual or not is provided, and the number of its symmetry axes is calculated. In chaps 4 and 5 all squared rectangles obtainable from one given c-net are computed. It is also determined whether the squared rectangle is perfect or imperfect and, in the latter case, whether the imperfection is trivial or not. Finally, chap. 6 shows a few typical results. In particular, we mention some typical output of the computer PASCAL and a list of squared squares of orders up to and including nineteen.

#### Résumé

Cette thèse examine les problèmes qui se posent lorsque le calcul des rectangles divisés en carrés est automatisé sur une calculatrice électronique. Après un chapitre 1 en guise d'introduction, le chapitre 2 montre comment un graphe complet est codifié de manière à pouvoir être traité par la calculatrice. Les propriétés du graphe par rapport à la question de savoir s'il est planaire sont particulièrement aisées à identifier en utilisant ce code. Il est montré comment, à partir du code établi pour le graphe original, on peut obtenir le code pour son dual. On y voit aussi comment, à partir du code du graphe, on peut trouver la matrice d'incidence aux sommets et les arrêtes du graphe. Au chapitre 3, un ensemble de codes  $\Sigma_{N+1}$ , représentant des graphes complets d'ordre N+1, est issu de l'ensemble des codes  $T_N$  de tous les différents graphes complets d'ordre N, en ajoutant une arête dans ces derniers graphes ou leurs duals. L'ensemble  $\Sigma_{N+1}$  peut contenir des codes représentant un seul et même graphe ou son dual. Pour cette raison, une méthode est décrite et montre comment un nombre peut être assigné pour chaque graphe complet, ce nombre caractérisant uniquement le graphe. Un classement effectué en tenant compte de ce nombre caractéristique donne l'ensemble des codes  $T_{N+1}$  de tous les différents graphes complets d'ordre N+1. Des informations supplémentaires quant à savoir si le graphe et son dual sont identiques ou non, sont fournies et le nombre de ses axes de symétrie y est calculé. Aux chapitres 4 et 5, tous les rectangles divisés en carrés pouvant être obtenus à partir d'un graphe donné sont calculés, de même que leurs codes Bouwkamp tels qu'ils seront imprimés par la calculatrice. Il y est aussi établi si la dissection est parfaite ou non, et dans ce dernier cas, si l'imperfection est triviale ou non. Enfin le chapitre 6 expose quelques résultats caractéristiques. Nous mentionnons particulièrement quelques réponses typiques données par la calculatrice PASCAL, ainsi qu'une liste des carrés divisés en carrés pour des valeurs jusqu'à et y compris dix-neuf.

#### Zusammenfassung

Diese Arbeit behandelt die bei der elektronischen Rechenmaschine bei der Automatisierung der Berechnung der in Quadrate unterteilten Rechtecke auftretenden Probleme. Nach der Einleitung in Kapitel 1 zeigt Kapitel 2 die Verschlüsselung eines sogenannten c-Netzes für die Verarbeitung in einer Rechenmaschine. Durch diese Verschlüssung lassen sich besonders leicht die Eigenschaften des Netzes, was seine Ebenheit betrifft, erkennen. Dann wird die Gewinnung der Verschlüsselung des Dualnetzes aus der Verschlüssung des Originalnetzes gezeigt. Auch wird beschrieben, wie sich die Knoten-Knoten Inzidenz Matrix und die Zweige des Netzes aus der Verschlüsselung des Netzes finden läßt. In Kapitel 3 wird eine Menge  $\Sigma_{N+1}$  von Verschlüsselungen, die c-Netze der (N+1)-ten Ordnung darstellen, aus der Menge  $T_N$  der Verschlüsse-lungen aller verschiedenen c-Netze N-ter Ordnung durch Hinzufügung eines Drahtes in den c-Netzen oder ihren Dualen gewonnen. Die Menge  $\Sigma_{N+1}$  kann Verschlüssungen enthalten, die ein und dasselbe Netz oder sein Dual darstellen. Es wird daher eine Methode angegeben, durch die sich jedem Netz eine Zahl zuordnen läßt, die das Netz eindeutig kennzeichnet. Das Sortieren nach dieser kennzeichnenden Zahl fürht zur Menge  $T_{N+1}$  der Verschlüsselungen aller verschiedenen c-Netze (N+1)ter Ordnung. Weiterhin wird angegeben, ob das Netz selbst-dual ist oder nicht. Es wird auch die Zahl seiner Symmetrieachsen berechnet. Im 4. und 5. Kapitel werden alle in Quadrate unterteilten Rechtecke für ein gegebenes c-Netz und ebenso ihre Bouwkamp-Verschlüsselungen, wie sie von der Rechenmaschine gedruckt werden, errechnet. Es wird auch bestimmt, ob das in Quadrate unterteilte Rechteck vollkommen oder unvollkommen ist. Im letzteren Fall wird festgestellt, ob die Unvollkommenheit trivial ist oder nicht. Kapitel 6 zeigt schließlich einige typische Beispiele. Im besonderen werden einige typische Ausgaben der Rechenmaschine PASCAL und eine Liste der in Quadrate unterteilten Quadrate bis zur neunzehnten Ordnung angeführt.

### CHAPTER 1

#### INTRODUCTION

This thesis is concerned with the problem of dissecting a rectangle into a finite number of non-overlapping squares. In particular, we study the problems that arise when one wants to calculate these dissections by an electronic computer.

The terminology of Brooks, Smith, Stone and Tutte<sup>1</sup>) and Bouwkamp<sup>2</sup>) will be used. A dissection of a rectangle into a finite number N > 1 of nonoverlapping squares is called a *squared rectangle* or a *squaring* of *order* N. The N squares are called the *elements* of the dissection. The term "elements" is also used for the (lengths of the) sides of the elements.

If the elements are all unequal, the squaring is called *perfect* and the rectangle is called a *perfect rectangle*; otherwise the squaring or rectangle is *imperfect*. Examples of perfect and imperfect squarings are given in figs 1 and 2; the numbers inscribed denote the lengths of the sides of the corresponding squares.

A squared rectangle that contains a smaller squared rectangle in its interior is called *compound*. All other squared rectangles are *simple*. Apparently, the squarings given in figs 1 and 2 are simple. An example of a compound squaring is given in fig. 3.

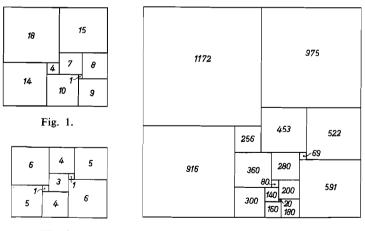


Fig. 2.

Fig. 3.

Fig. 1. Example of a perfect squaring of order 9.

Fig. 2. Example of an imperfect squaring of order 9.

Fig. 3. Example of a compound perfect squaring of order 17.

The squaring is called *trivially imperfect* if it contains equal elements that touch each other along a common side.

In 1903 Dehn<sup>3</sup>) initiated the study of a somewhat more general problem, namely, that of the (non-trivial) dissection of a rectangle into a finite number of non-overlapping smaller rectangles. He proved the following theorem: If each sub-rectangle has commensurable sides, then so has the original rectangle and, moreover, all the sides of the rectangle and the sub-rectangles are mutually commensurable.

In particular, by taking the sub-rectangles to be squares, Dehn found as a corollary: Any squared rectangle has commensurable sides and elements.

Dehn did not go beyond announcing (and proving) this theorem and its corollary. It remained an open question whether a perfect squared rectangle did exist at all. However, such a squared rectangle was given in 1925 by Moron<sup>4</sup>), when he gave the example of fig. 1.

Considerable progress was made by Brooks, Smith, Stone and Tutte <sup>1</sup>) in 1940. They succeeded in separating the topological part from the metrical part of the problem. The topological part of the problem appeared to be related to the theory of linear graphs, while the metrical part proved to be connected to the theory of current flow in electrical networks. They also gave a short table of low-order squared rectangles.

The relation of the squared rectangles with electrical networks was also considered by Bouwkamp<sup>2</sup>) who gave a more-physical approach to the problem. In Bouwkamp's paper a table was given of all simple squared rectangles of orders up to and including 13. For that purpose, Bouwkamp introduced a concise and efficient code for the squared rectangle. He supposed the rectangle to be drawn in such a manner that its larger side is horizontal. Further, the element at the upper-left corner should not be smaller than the three remaining corner elements. After this orientation of the rectangle, the upper-left corner of each element is taken as its representative point. The length of the sides of the elements for which the representative points lie in the same horizontal segment are assembled within parentheses in the order from left to right, the elements within parentheses being separated by commas. The parentheses read in order from top to bottom of the rectangle. Collinear horizontal segments are taken in order from left to right. This code will be called the Bouwkamp code of the squaring. For example, the codes pertaining to the squarings given in figs 1 and 2 are as follows:

### (18,15)(7,8)(14,4)(10,1)(9) and (6,4,5)(3,1)(6)(5,1)(4).

Brooks, Smith, Stone and Tutte <sup>1</sup>) proved that there are no perfect rectangles of order less than 9. The minimum number of elements necessary to divide a square simply is also known <sup>2</sup>). It is a simple imperfect squared square of order 13. Its code reads as follows: (12,11)(1,3,7)(11,2)(5)(2,5)(4,1)(3). Other examples

of simple imperfect squares were found by Bouwkamp, Duijvestijn and Medema <sup>5</sup>). There are none of order 14; those of order 15 are:

> (20,8,11)(5,3)(2,12)(7)(19,8)(5,7)(11,2)(9),(20,19)(1,3,8,7)(19,2)(5)(2,5)(12,1)(3)(8),(23,18)(7,11)(18,3,2)(1,5,3)(4)(2,1)(12)(11).

Simple squared squares of higher order are given in chap. 6.

Willcocks <sup>3</sup>) has constructed a perfect squared square of order 24, with code as follows:

(81,51,43)(8,35)(30,29)(2,33)(31)(39,14,20,38)(9,5)(4,1)(3,18)(16)(64)(56)(55).

However, this square is compound in that it is built up of one square and two squared rectangles. At present it is not known whether 24 is the minimum number of elements necessary to divide a square perfectly. As to perfect and simple squares, the best result known sofar is also due to Willcocks <sup>7</sup>), who has found a simple perfect square of order 37, with code as follows:

 $(728,378,406,435)(350,28)(405,29)(464)(648,347,83)(184,206,98)(10,454)(108)\\(162,22)(336)(245,102)(20,142)(122)(210,54)(56,189)(250,594)(571,133)(438,94)\\(344).$ 

The existing tables of low-order squarings have been useful for the construction of squarings of special type (cf. the 24-order squared square of Willcocks). For that reason, Ellis <sup>8</sup>) started to extend the tables of perfect squarings so as to include those of order 14. These calculations were entirely done by hand, that is, without the use of electronic calculating machinery. It is practically impossible to continue in this way to orders 15 and higher. Further extension can only be carried out with electronic computation.

In trying to solve the problem of generating squared rectangles automatically with an electronic computer, one meets a number of new problems. Especially, how can the computer deal with the topolocigal aspects of the problem?

In the present thesis it will be described how the necessary new networks can be generated automatically. A characteristic of the network will be calculated, by which it can be judged whether two networks are different or the same. Furthermore, it will be described how the Bouwkamp codes of all dissections arising from a given network can be obtained automatically; it is also possible to let the computer indicate whether a squaring is perfect or imperfect, and in the latter case whether the imperfection is trivial or not. The first results have been published by Bouwkamp, Duijvestijn and Medema 5)<sup>9</sup>), where all simple squarings of orders up to and including 15 were given.

In describing the programmes occurring in this paper we closely followed the rules of the ALGOL-60 language <sup>10</sup>). In the ALGOL programmes it is assumed that non-local variables have been introduced previously, unless stated other-

wise. The programmes written in ALGOL were translated into PASCAL (Philips automatic sequence calculator) code. With the aid of these programmes all networks of orders up to and including 19 were generated on PASCAL. Further, for all possible networks of orders up to and including 20, possible squared squares following from these were determined; the Bouwkamp codes of these squared squares were printed by PASCAL.

Some of the results will be given in chap. 6. For example, in contrast with early expectation, there does not exist a simple perfect square of order less than 20.

### CHAPTER 2

#### **PROBLEMS OF CODING**

#### 2.1. Introduction

The relation of squared rectangles to planar electrical networks will now be considered. It was shown in papers already referred to  $1)^2$ ) that each element of the squared rectangle corresponds to a wire or branch, while each horizontal line segment corresponds to a vertex, and each vertical line segment to a mesh not containing other parts of the network in its interior. The vertices corresponding to the upper and lower horizontal sides are the poles of the network.

The network constructed in this way is called the *normal polar network* or *normal p-net*  $^{1}$ ; see the example of fig. 4.

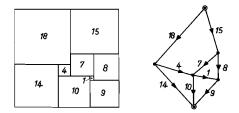


Fig. 4. Example of a squaring and its associated normal p-net;  $\odot$  = pole of the net.

A planar network (with more than one vertex) that is connected is called a *net*. If two vertices of the net are assigned as poles, and no circuit is enclosing the poles, the net is called a *polar net* or a *p-net*.

A *c-net* is a net that has no parts (consisting of more than one wire, and less than all but one wire) joined to the rest at less than three vertices. Joining the poles of a normal p-net by a wire gives a c-net (c) if the squaring corresponding to the normal p-net is simple.

Now before the normal p-net is constructed, the rectangle can be turned through 90 degrees. Then joining the two poles by a wire again produces a c-net (c'). The net c' is called the *dual* of the net c. Obviously, c is also the dual of c'. Therefore, c and c' form a pair of dual c-nets.

Dual nets can be drawn in such a way that the vertices of either of them lie inside the corresponding meshes of the other, while corresponding branches, and only these, cross each other. Brooks, Smith, Stone and Tutte<sup>1</sup>) proved that the dual of a c-net is a c-net. Apparently, any simple squaring can be obtained from an appropriate c-net.

To illustrate the various concepts, consider the c-net of fig. 5, obtained from the normal polar net of fig. 4. For later purposes this c-net will be called the *reference net*. The reference net and its dual (dashed lines) are drawn in fig. 6.

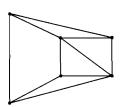


Fig. 5. The reference net.

First some notations will be introduced. The number of vertices of a net will be denoted by K, that of its dual by K', while the number of meshes is denoted by M and M' respectively. The number of wires is denoted by B. Henceforth B is called the *order of the net*. Apparently one has M' = K, K' = M, while according to the theorem of Euler the following relation holds:

$$K+M=K'+M'=B+2.$$

Let N be a net with vertices  $V_1, \ldots, V_K, K \ge 2$ , and let INC[i, j] be a matrix such that

INC 
$$[i, j] = 0$$
, if  $V_i$  and  $V_j$  are not connected,  
INC  $[i, j] = -1$ , if  $V_i$  and  $V_j$  are connected,  
INC  $[i, i] =$  the number of wires at  $V_i$ .

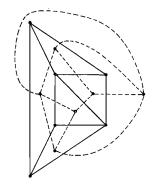


Fig. 6. The reference net and its dual.

It was shown by Brooks, Smith, Stone and Tutte <sup>1</sup>) that all first cofactors of INC are the same, except for the sign. Their common absolute value is called the *complexity* of the net; it is denoted by C. It can be shown that dual nets have equal complexities; furthermore the complexity equals the number of complete trees of the net <sup>1</sup>).

Simple squarings can be obtained from a c-net by placing an electromotive force of value C in one of the wires (all wires have unit resistance). The current flow caused in the network is called the *full flow*, while the currents are called the *full currents*. The highest common factor (HCF) of the full currents is

called the *reduction factor*, denoted by RF. If instead an electromotive force of value C/(HCF) is placed in the wire under consideration, one obtains the *reduced flow* and the *reduced currents*.

The sides of the squaring obtained in this way are the *full sides* and *reduced* sides respectively. The full horizontal side equals the current caused by an electromotive force of value C in its own wire, while the full vertical side is equal to the potential difference between the two ends. The reduced horizontal and reduced vertical sides are obtained if instead an electromotive force of value C/(HCF) is applied.

A squared rectangle that contains a squared rectangle of lower order in its interior and any corresponding p-net are called *compound*; all other squared rectangles and p-nets are *simple*. If a p-net has a part not containing a pole joined to the rest by only two wires, or if it has a pair of vertices joined by two (or more) wires, then these wires will have equal currents. If these currents are not zero, the resulting imperfection is said to be *trivial*.

#### 2.2. Code of the c-net

Next we come to the question of how a general network can be stored into an electronic computer. Obviously the vertex-vertex incidence matrix INC can be used for this purpose; the network is determined uniquely by the matrix INC, and vice versa.

However, it is quite difficult to find out whether the network is planar or not if only the matrix INC is given. In addition, even if the network is known to be planar, it is still difficult to draw the net without crossings from the knowledge of INC alone.

In order to overcome these difficulties, a new code is introduced. It is assumed that the planar network is drawn on the sphere. The vertices are numbered arbitrarily from 1 to K.

The boundary of a mesh contains a set of vertices. A code of a mesh is obtained as follows: While walking in the positive sense along the boundary of the mesh, starting with  $V_i$ , we encounter  $V_j$ ,  $V_k$ ,  $V_l$ , ..., until we return to  $V_i$ . The sequence  $V_i$ ,  $V_j$ ,  $V_k$ , ...,  $V_i$  is a code of the mesh.

#### Example:

A possible code of mesh 1 of the reference net is 12651, as can be seen from fig. 7; but we can also take 26512, 65126 or 51265.

A code of a net is the sequence of codes of all its meshes separated by zeros. At the end two more zeros are added. Hence this code of the net can be considered as a vector V[t], t = 1, 2, ..., 2(B + M) + 1.

#### Example:

A code of the reference net is as follows:

1 2 6 5 1 0 2 3 6 2 0 3 5 6 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0.

It should be noticed that a different code would have been obtained if the vertices were enumerated in another way. Furthermore, the chosen codes of the meshes may be permuted in the code considered. Any of the codes so obtained is sufficient to characterize the net topologically.

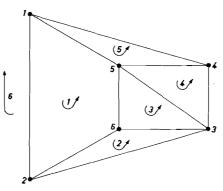


Fig. 7. The reference net.

#### 2.3. Determination of the branches of the c-net

A wire contains two vertices of the net. To each pair of vertices  $V_i$  and  $V_j$  of a wire two arrows are associated. The first is directed from  $V_i$  to  $V_j$  and the second from  $V_j$  to  $V_i$ . The wire with the arrow directed from  $V_i$  to  $V_j$  will be called *branch*  $V_iV_j$ , the other is *branch*  $V_jV_i$ . A branch is therefore an oriented wire; it contains two vertices, which are denoted by *branch* 1 and *branch* 2. If only one of the two branches  $V_iV_j$  and  $V_jV_i$  is used to indicate the associated wire, then the net has B branches.

Therefore the branch *i* is denoted by its two vertices, namely, branch 1[i] and branch 2[i], with i = 1, 2, ..., B. In the same way the branches of the dual net are denoted by branchdual 1[i] and branchdual 2[i]. It is further assumed that the meshes are numbered from 1 to M in the same sequence as their codes occur in the code of the net.

The branches of the net and its dual can be derived from the code V[t] of the net by the following programme:

procedure form branches (V, branch 1, branch 2, branchdual 1, branchdual 2,

K, M);

integer K, M;

integer array branch 1, branch 2, branchdual 1, branchdual 2, V; begin integer m, t, tt, i;

t:=m:=1; tt:=0;

```
begin: for i := 1 step 1 until tt do
```

begin

if  $V[t+1] = \text{branch } 1[i] \land V[t] = \text{branch } 2[i]$ 

then begin branchdual 2[i] := m; go to next end end i: tt: = tt + 1; branch 1[tt]: = V[t]; branch 2[tt]: = V[t+1]; branchdual 1[tt] := m; next: t := t + 1; if V[t+1] = 0then begin if V[t+2] = 0then go to end; m: = m + 1; t: = t + 2end; go to begin; end: B: = tt; M: = m; K: = B + 2 - Mend form branches

### Example:

Applying the **procedure** form branches to the code of the reference net one obtains a set of branches which are given below:

i	branch 1[i]	branch 2[i]	branchdual 1[i]	branchdual 2[i]
1	1	2	1	6
2	2	6	1	2
3	6	5	1	3
4	5	1	1	5
5	2	3	2	6
6	3	6	2	3
7	3	5	3	4
8	3	4	4	6
9	4	5	4	5
10	4	1	5	6

### 2.4. Dualization of the c-net

From the code V[t] of the net it is possible to obtain the code of the dual net. To see this, it is first necessary to define the concept of adjacent vertex and adjacent branch of a vertex. An *adjacent vertex*  $V_2$  of a vertex  $V_1$  is a vertex that is connected to  $V_1$ . The branch  $V_1V_2$  will be called an *adjacent branch* of  $V_1$ .

A mesh is said to be *left* of a branch  $V_iV_j$  of its boundary, if the sequence  $V_iV_j$  occurs in the code of the mesh. In that case the branch is said to be right of the

mesh. The mesh is said to be *right* of a branch  $V_iV_j$  if the sequence  $V_jV_i$  occurs in the code of the mesh; if so the branch is left of the mesh.

### Example:

In the reference net the mesh 1 is left of branch 12, but right of branch 21.

Now it is known that the vertices of the dual net are corresponding to the meshes of the original net and that the meshes of the dual net are corresponding to the vertices of the original net. Assuming again that the vertices are numbered from 1 to K, while the meshes are numbered from 1 to M, according to the occurrence of their codes in the code of the net, we choose the enumeration of the meshes of the dual net equal to the enumeration of their corresponding vertices of the original net. The same is done for the vertices of the dual net and the meshes of the original net.

Next we consider an arbitrary vertex  $V_0$ . To this vertex  $V_0$  a set of *left-cyclic-ordered adjacent branches* will be associated in the following way: Take an arbitrary adjacent vertex  $V_1$  of  $V_0$ ; then search the *left* mesh  $L_1$  of  $V_0V_1$ ; then search the other adjacent vertex  $V_2$  of  $V_0$  in  $L_1$ ; then search the *left* mesh  $L_2$  of  $V_0V_2$ ; and so on, until  $V_1$  has been reached again.

The set  $V_0V_1$ ,  $V_0V_2$ , ...,  $V_0V_k$ , where  $k \ge 3$ , will be called the left-cyclicordered adjacent branches of  $V_0$ . If *left* is replaced by *right*, then the rightcyclic-ordered adjacent branches of  $V_0$  are obtained.

The sequence  $L_1, L_2, \ldots, L_k, L_1$  is precisely a code of the mesh  $V_0$  of the dual net.

If this procedure is carried out for all vertices of the net, the code of the dual net is obtained. How it can be done in an automatic way is described by the **procedure** dualize, as follows:

procedure dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, V);

```
integer K;
```

integer array branch 1, branch 2, branchdual 1, branchdual 2, V; begin integer *i*, *j*, *l*, *h*, *t*, search, remember;

```
integer array vector 1, vector 2[1:B];

t: = 0; i: = 1;

start: l: = 1;

for j: = 1 step 1 until B do

begin

if branch 1[j] = i

then

begin

vector 2[l]: = branchdual 1[j];

vector 1[l]: = branchdual 2[j]; l: = l + 1

end;

if branch 2[j] = i
```

then begin vector 1[l]: = branchdual 1[j]; vector 2[l]: = branchdual 2[j]; l: = l + 1end end; t: = t + 1; V[t]: = vector 1[1]; search: = remember: = vector 2[1]; begin: for h := 1 step 1 until l-1 do begin if vector 1[h] = searchthen begin t: = t + 1; V[t]: = search; search: = vector 2[h]; if search = remember then go to continue; go to begin end end: continue: t := t + 1; V[t] := 0; i := i + 1; if i = K + 1then go to end;

go to start; end: t := t + 1; V[t] := 0end dualize

### Example:

The code of the dual of the reference net will become 6 1 5 6 0 1 6 2 1 0 2 6 4 3 2 0 4 6 5 4 0 1 3 4 5 1 0 1 2 3 1 0 0.

### 2.5. Determination of the vertex-vertex incidence matrix of the c-net

The vertex-vertex incidence matrix INC is easily determined from the branches of the original net. It is described by the following programme: **begin integer** i, j, k, m, n;

end

### 2.6. Wheels

Finally, a set of special nets are worth mentioning, namely the so-called wheels. A *wheel* is a c-net with an even number B of branches, with one vertex  $p_{1B}$  and B-1 vertices  $p_3$ , where  $p_k$  means a vertex incident with k branches (see fig. 8).

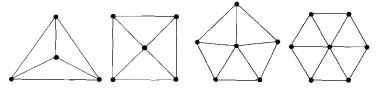


Fig. 8. The first few low-order wheels.

The code of a wheel having B branches is determined by the **procedure** wheel. It is supposed that W has been declared as **integer array** variable.

```
procedure wheel (B); value B;
         integer B;
begin integer MDP, t, l;
     MDP: = B \div 2 +1; t: = 1;
     for l := 1 step 1 until MDP-2 do
        begin
             W[t]: = W[t+3]: = l; W[t+1]: = MDP; W[t+2]: = l+1;
             W[t+4]:=0; t:=t+5
        end:
     W[t]: = W[t+3]: = MDP - 1; W[t+1]: = MDP; W[t+2]: = 1;
     W[t+4]:=0; t:=t+5;
     for l := 1 step 1 until MDP - 1 do
        begin
             W[t] := l; t := t + 1
        end;
     W[t]:=1; W[t+1]:=W[t+2]:=0
end wheel
```

### CHAPTER 3

### **IDENTIFICATION PROBLEM**

#### 3.1. Introduction

Consider the set  $S_B$  of c-nets having B wires. Let s be an element of  $S_B$  and s' its dual. Then, according to Tutte <sup>11</sup>), we have the following theorem: If s is not a wheel, then at least one of the nets s and s' can be constructed from an element  $\sigma$  of  $S_{B-1}$  by addition of a wire joining two vertices of  $\sigma$ .

With the aid of this theorem, the set  $S_B$  can be constructed from the set  $S_{B-1}$ . To this end, we start with the set  $T_{B-1}$  of the codes of the (B-1)-wire c-nets; for each element of  $S_{B-1}$ , we have one element of  $T_{B-1}$ . Take one element of the set  $T_{B-1}$ ; it represents a certain (B-1)-wire c-net. Add a wire in this c-net, in so far as the result is a *B*-wire c-net, and construct a code representing the latter c-net. If this procedure is carried out for all elements of  $T_{B-1}$  and for all possibilities of adding wires, a set  $\Sigma_B$  of codes is obtained, of which each code represents a *B*-wire c-net. Let *s* be an element of  $S_B$ , then either *s* or its dual *s'* is represented by an element of  $\Sigma_B$ .

In the set  $\Sigma_B$  there may be many codes representing the same net. Now two questions arise:

- (1) How can the set  $\Sigma_B$  be constructed in an automatic way on an electronic computer?
- (2) How (if  $\Sigma_B$  is available) can equal nets represented by different codes be identified, and how can this be done on a computer, so as to obtain the set  $T_B$ ?

#### 3.2. Generation of nets by means of their codes

Apparently, addition of a wire to a c-net s, by joining two of its vertices, gives a non-planar network unless these two vertices belong (before joining) to one and the same mesh of s.

Let the net s contain a mesh R of b wires (b > 3) and let  $V_1 V_2 V_3 \dots V_b V_1$ be the code of R. Apparently a net  $s^*$  is obtained if two vertices  $V_i$  and  $V_j$ , not being adjacent vertices, are joined by a wire. This can be done in b(b-3)/2different ways, and in any of these ways the mesh R is split into two smaller meshes,  $R_1$  and  $R_2$ .

For example, if  $V_1$  is joined to  $V_3$  two new meshes having the following codes are obtained:  $V_1 V_2 V_3 V_1$  and  $V_3 V_4 \ldots V_b V_1 V_3$ . The total number of elements of these codes exceeds the number of elements in the code of mesh R by 3. This is true if two arbitrary non-adjacent vertices of R are joined, for any R of s. Hence the number of elements of the code of the new net  $s^*$  exceeds the number of elements of the code of the original net s by 4 (in the code of  $s^*$ , the codes of  $R_1$  and  $R_2$  are separated by the element 0). In the following programme it is described how, starting from a code representing a certain net, the codes of the new nets are obtained (addition of a wire in the original net). If the original net is not selfdual, then the dual net is constructed and the procedure repeated (addition of a wire in the dual net).

```
procedure generate nets (W); integer array W;
          comment if the net from which the new nets are generated is selfdual,
                    it is assumed that a Boolean variable selfdual is true, other
                    wise selfdual is false;
begin
        Boolean dualized;
        dualized: = false; go to con 2;
con 1: if dualized \vee selfdual
        then go to finished;
        form branches (W, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
       dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, W);
          dualized: = true:
con 2:
          begin integer i, ii, m, s, t, MM, p, q, a, b, l;
              integer array sum [1: M + 1], multiplicity [1: M];
              m: = t: = \text{sum } [1]: = i: = 1;
        label: if W[t+2] = 0
              then
                  begin
                        t := t + 3; i := i + 1; sum [i] := t;
                        multiplicity [i-1]: = m; m: = 1; if W[t] = 0
                                                          then go to follow
                  end:
              t := t + 1; m := m + 1; go to label;
       follow: MM := i - 1;
              for ii := 1 step 1 until MM do
                  begin
                       if multiplicity [ii] > 3
                       then
          begin
            q:= sum [ii] - 1;
            for a: = 1 step 1 until sum [ii] - 1 do V[a]: = W[a];
            for b := sum [ii + 1] step 1 until sum [MM+1] do V[b+4] := W[b];
            for s: = 1 step 1 until multiplicity [ii] - 2 do
                begin
                  for l := s + 2 step 1 until if s = 1 then multiplicity [ii] - 1
                                                      else multiplicity [ii] do
                    begin
                      p:=q+1;
```

```
for m: = s step 1 until l do
       begin
          V[p]: = W[m+q]; p: = p+1
       end:
    V[p]: = W[s+q]; p: = p+1; V[p]: = 0; p: = p+1;
         m := l step 1 until multiplicity [ii] do
    for
          begin
           V[p]: W[m+q]; p:=p+1
         end:
    for
         m: = 1 step 1 until s do
         begin
           V[p]: = W[m+q]; p: = p+1
         end;
    V[p]: = W[l+q]; V[p+1]: =0;
       comment at this point the net can be identified,
       the procedure form TNSTAR will be explained
       later: form TNSTAR:
     end l
end s
```

end if

end *ii* 

end block con 2; go to con 1;

finished:

end generate net

### Example:

From the code of the reference net (which is selfdual) four new codes can be generated. The new codes are denoted by  $V_k[t]$  (k=1,2,3,4), while that of the reference net is denoted by W[t].

t = 1(1)37 W[t] = 1 2 6 5 1 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0  $V_1[t] = 1 2 6 1 0 6 5 1 6 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0$   $V_2[t] = 2 6 5 2 0 5 1 2 5 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0$   $V_3[t] = 1 2 6 5 1 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0$   $V_3[t] = 1 2 6 5 1 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 3 0 0$   $V_3[t] = 1 2 6 5 1 0 2 3 6 2 0 3 6 5 3 0 3 4 5 3 0 1 5 4 1 0 1 4 3 2 1 0 0$ 

### 3.3. Identification problem

We now return to question (2) of sec. 3.1, which may be phrased somewhat differently as follows: How can we find out whether or not two different codes represent one and the same net? What is more, how can we uniquely characterize the net if and when it is represented by one of its many possible codes? This set of problems is henceforth referred to by the expression "identification problem". To solve this identification problem is of course much more complicated than the construction of the set of codes  $\Sigma_B$ .

Two nets are equal if an enumeration of the vertices can be found such that the vertex-vertex incidence matrices INC of the two nets are equal. In principle it is possible to run through all K! permutations of the vertices of one net and compare the corresponding incidence matrices with that of the other net. However, such a procedure takes a long time, even on a fast computer.

It would be much better if from the code there could be found a characteristic of the net determining the latter in a unique way. In a first attempt to find such a characteristic, we tried several simple and obvious possibilities. However, already at an early stage it became apparent that these characteristics did fail to characterize the net uniquely.

The characteristics can be divided into two types: Type 1 of characteristic is such that two nets having different characteristics are different. Type 2 of characteristic is such that two nets having equal characteristics are equal. Apparently a characteristic of both type 1 and type 2 determines the net uniquely.

#### 3.4. Type-1 characteristics

We will see in how far the identification problem can be solved if use is made of a characteristic that is of type 1 and not of type 2. If the generation process of sec. 3.2 is applied to the set  $T_{B-1}$  the set  $\Sigma_B$  is obtained. The set  $\Sigma_B$  has many more elements than the set  $T_B$ . That means, many nets corresponding to codes in  $\Sigma_B$  are equal. With the characteristic under consideration, nets having different characteristics can be discriminated. However, nets having equal characteristics need not be equal; that is, the remaining undiscriminated nets represented by elements of  $\Sigma_B$  have to be tested in a different way. This causes much extra labour if the set  $\Sigma_B$  is much larger than the set  $T_B$ .

Some simple examples of characteristics of type 1 will be discussed now. The first example is a vector A of which the elements A[k] denote the number of vertices incident with k wires,  $k \ge 3$ . The reference net consists of 2  $p_4$ 's and 4  $p_3$ 's. Hence A[3] = 4 and A[4] = 2. A short notation is  $A = 3^{4}4^2$ .

Another example is the combination of the characteristic A of a net with A' of its dual, (A, A'). For example, in the case of the reference net we have  $(3^{4}4^{2}, 3^{4}4^{2})$ .

That the characteristic (A,A') is not of type 2 can be seen from fig. 9, where two different nets with the same characteristic (A,A') are shown.

A third and last example of characteristics of type 1 is due to Bouwkamp. He considered a matrix D of which the elements D[k,j] denote the number of wires that connect a vertex  $p_k$  to a vertex  $p_j$ . Apparently D is symmetric. Furthermore, D has the property that the sum of the elements to the right of the main diagonal plus the trace equals the number of wires B of the net. For the reference net and its dual the matrices D and D' are as follows:

$$D = \left\| \begin{array}{cc} 3 & 6 \\ 6 & 1 \end{array} \right\|, \qquad D' = \left\| \begin{array}{cc} 3 & 6 \\ 6 & 1 \end{array} \right\|.$$

The combination of the four characteristics A,A',D,D' will be denoted by I = (A,A',D,D'). It is easy for the computer to determine I from the code of the net, but I is by no means fully discriminating. For example,  $S_{16}$  has 249 elements, except for duals, but there are in this case only 169 different characteristics I (see also fig. 9).

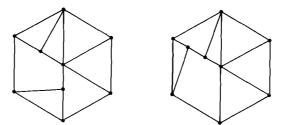


Fig. 9. Example of two distinct c-nets with the same characteristic I.

#### 3.5. Type-2 characteristics

A characteristic of type 2 can be used as a sieve. Nets having equal characteristics can be omitted. Especially if the characteristic is selective not much extra work has to be done. First the remaining nets having different characteristics can be classified according to their complexity. Only nets having equal complexities have to be investigated. Now the characteristic I of type 1 can be applied. If this does not discriminate either, then at last the Bouwkamp codes can be calculated; with the aid of these codes two nets can always be discriminated.

#### 3.6. A characteristic of both type 1 and type 2

Consider the vertex-vertex incidence matrix INC of the net as obtained from the code of the latter. The off-diagonal elements of INC are either zero or minus one. We replace the off-diagonal elements by their absolute values. Then an element on the diagonal is the sum of the off-diagonal elements in the same row (or column). The new matrix will be denoted by X, with elements  $X_{ij}$  (i, j = 1, 2, ..., K).

To X an integer G(X) will be associated. The binary notation of G(X) is obtained by writing the elements of X to the right of the main diagonal in the sequence  $X_{12} X_{13} \ldots X_{1K} X_{23} X_{24} \ldots X_{2K} \ldots X_{K-1,K}$  so that its decimal value is given by

$$G(X) = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} X_{ij} \ 2^{\frac{1}{2}(K-i)(K-i-1)+K-j}.$$

G(X) is called the identification number of the net in relation to the code of the net under consideration. If G(X) is known, X and INC are known, and vice versa: From the identification number the upper triangle of X can be constructed while the lower triangle follows from the symmetry of X; the diagonal elements may be found from the sums of the off-diagonal elements in the same row.

Let now the matrix X be transformed by interchanging the kth row with the lth row and at the same time the kth column with the lth column. This transformation is nothing but a new enumeration of the vertices; such an enumeration is called a permutation. For every permutation we have an X and the corresponding G(X). Let GM be the maximum of G(X) on the group of permutations. The number GM is independent of the particular choice of the code of the net. Hence GM is a characteristic of both type 1 and type 2; it is called *identification magnitude*.

A permutation (there may be more than one) for which G(X) is maximum on the group of permutations of X brings the matrix in the maximal form, say. The matrix can be brought into this maximal form by running through all possible permutations (there are K! of them) and by testing which permutation gives the maximal G. If K is large this process is time consuming.

Instead of considering the full permutation group one can consider a subgroup of the group of all permutations (by imposing enough requirements on X) and maximize G(X) on this subgroup.

Let p be a permutation of the full permutation group of the net. To each p there corresponds an identification number  $G(X_p)$ . If  $G(X_{p1}) = G(X_{p2}) = \ldots$ =  $G(X_{pi})$  we identify the elements  $p_1, \ldots, p_i$  to an element h. These new elements h form the set H.

If there exist  $p_1$  and  $p_2$  such that  $G(Xp_1) = G(Xp_2)$  it is possible to deform the net topologically, after having fixed the enumeration (corresponding to the permutation  $p_1$ ) to the vertices, such that the deformed net can be considered as the non-deformed net with an enumeration corresponding to the permutation  $p_2$ .

Next, let the set  $H^* \subset H$  be such that if  $h^* \in H^*$  the permutations corresponding to  $h^*$  are satisfying certain criteria  $CR_1, CR_2, \ldots, CR_s$ . Then the identification problem is solved if enough criteria can be found (i.e. *s* just so large) that the set  $H^*$  contains only one element. If  $H^*$  contains more than one element the identification number *G* can be maximized on  $H^*$ , and the work involved may be considerably less compared to the maximization on the full permutation group.

Another possibility to determine the maximum of G on a certain permutation group  $H^{**}$  is to construct certain paths through  $H^{**}$ , of which it is known that they lead to the maximal G on  $H^{**}$  (steepest ascent).

### 3.7. Example of a type-2 characteristic

Instead of maximizing the identification number G one can maximize other

numbers defined on the permutation group. For example, the following procedure was attempted. In a maximalization process tested on one of the available computers the number  $G^*$  was maximized where  $G^*$  is defined by

$$G^* = \sum_{i=1}^{K-1} \sum_{j=i+1}^{K} X_{ij} \{ 2^{(K-j)(K-1)+K-i-1} + 2^{(K-i)(K-1)+K-j} \}.$$

It is assumed that the main diagonal elements of the matrix X are non-increasing and remain so in the maximalization process. The transformation applied to Xwas the interchange of two rows and the corresponding columns. The columns *j* and k (and the corresponding rows) were tried for interchange when X[i,j] = 0and X[i,k] = 1, k > j, while the main diagonal elements remained non-increasing. The process was stopped when no i, j and k could be found such that  $G^*$  increased when the columns *i* and *k* were interchanged. The reason why this process works only as a sieve is that there are cases where more than two rows and columns have to be interchanged simultaneously in order to increase  $G^*$ . The sieve works much better if the method is applied to both the original net and its dual. This was tested on those codes of  $\Sigma_{16}$  that are representing nets for which K = M = 7 and it gave perfect discrimination. The method is still unsatisfactory even when both the original and dual nets are "maximized" because a special programme is necessary for identifying the nets as soon as the identification numbers corresponding to the "maximum" permutation have been calculated: one has also to remember which nets are dual. The method that can be used is that of drawing chains in the set  $\Sigma_B$ . A chain can be drawn either when two codes correspond to nets having equal identification numbers or when it is known that the nets are dual. The process of drawing chains has been carried out on a computer.

#### 3.8. Weights and scores

With the method of "weights and scores" a sequence of importance of the vertices of the net is calculated that does not depend on the particular code representing the net. As soon as a sequence of importance (this is a permutation) is known the identification number corresponding to that permutation is calculated. This identification number is used to characterize the net.

To each vertex of the net a weight is assigned; all weights are assembled in a vector: weight [i], i = 1(1)K. The weights can change during the process; the process of weights and scores is ready when the weights of all vertices are different.

The process starts with the weights of all vertices equal to 2. New weights are assigned after "scores" have been calculated. The scores are given by a vector: score [i], i = 1(1)K. Depending on the value of a **Boolean** variable: fromdual, scores are calculated with the aid of the weights of the original or the dual net. The scores are calculated by the following programme:

```
begin integer i;
      Boolean fromdual;
      integer array weight original, score [1:K], weight dual [1:M];
      for i := 1 step 1 until K do score [i] := 0;
      if - fromdual
      then
       for i := 1 step 1 until B do
       begin
       score [branch 1 [i]]: = score [branch 1 [i]] + weight original [branch 2 [i]];
       score [branch 2[i]]: = score [branch 2[i]] + weight original [branch 1[i]]
       end
      else
       for i := 1 step 1 until B do
       begin
       score [branch 1[i]]: = score [branch 1[i]]+weight dual [branchdual 2[i]];
       score [branch 2[i]]: = score [branch 2[i]]+weight dual [branchdual 1[i]]
       end
```

### end

Example:

For the reference net the start is as follows:

ı,

vertex i	weight [i]
1	2
2	2
3	2
4	2
5	2
6	2

When the scores are calculated (fromdual is false) one obtains

i	score [i]
1	6
2	6
3	8
4	6
5	8
6	6

With the aid of the score a new weight is calculated. One tries to discriminate between vertices that have equal weights so far, by means of their scores. First, all vertices of weight 2 are searched; those among them having the lowest score get a new weight equal to 2. The vertices having the next lowest score get a new weight twice as large, and so on. Then all vertices of (old) weight 4 are searched;

those among them having the lowest score get a new weight twice as large as the last given weight, and so on, until all vertices have got new weights.

The maximum weight is remembered. Then again new scores are calculated. The process is stopped if the maximum weight assigned equals  $2^{\kappa}$  or if the maximum weight has not been changed. In the latter case we shall say there is "no gain". Consequently, in continuing the discussion of our example, the next step for the reference net is

i	weight [i]	score [i]
1	2	8
2	2	8
3	4	10
4	2	10
5	4	10
6	2	10
Maximum weight $= 4$ .		
Then		
i	weight [i]	score [i]
1	2	14
2	2	14
3	8	18
4	4	18
5	8	18
6	4	18
Maximum weight $= 8$ .		
Finally		
i	weight [i]	
1	2	
2	2	
3	8	
4	4	
5	8	
6	4	

Maximum weight = 8, hence there is no gain. The reason that the process stops on no gain, if it is applied to the reference net, is that the reference net has certain symmetry properties. Apparently in any permutation of the reference net the vertices 1 and 2, 4 and 6, 3 and 5 can be interchanged without changing the identification number corresponding to that permutation.

### 3.9. Procedure identify

Now the **procedure** identify will be described. When the branches of the original and dual nets have been found, this procedure calculates a sequence of importance of the vertices. This sequence is given by a vector: location [i], i = 1(1)K. If, entering the procedure, fromdual is **false**, scores are calculated with the aid of the original net alone. If fromdual is **true**, scores are calculated the first time with the aid of the dual net, and later with the aid of the original net. If, after coming back from the procedure, the **Boolean** variable nogain is **false**, the identification process is ready; if nogain is **true**, the identification process is not yet ready. The maximum weight assigned is indicated by the procedure. It is assumed that weight original [K+1] is equal to zero. The **procedure** identify is given by the following programme:

```
procedure identify (K, weight original, weight dual, location, maxweight, <math>n,
           branch 1, branch 2, branchdual 1, branchdual 2);
integer K, n, maxweight;
integer array weight original, weight dual, location, branch 1, branch 2, branch
              dual 1, branchdual 2;
begin integer z, weightstorage, t, i, k, q, s, l, min;
      Boolean ready;
      integer array new location, score [1:K];
      nogain: = false;
start: z := 1; weightstorage: = 1; t := 1;
      for i := 1 step 1 until K do score [i] := 0;
      if fromdual
      then
           for i := 1 step 1 until B do
           begin
             score [branch 1 [i]]: = score [branch 1 [i]] + weight dual [branch
             dual 2[i]];
             score [branch 2 [i]]: = score [branch 2 [i]] + weight dual [branch
             dual 1 [i]];
             fromdual: = false
           end
      else
           for i := 1 step 1 until B do
            begin
             score [branch 1 [i]]: = score [branch 1 [i]] + weight original
             [branch 2 [i]];
             score [branch 2 [i]]: = score [branch 2 [i]] + weight original
             [branch 1 [i]];
            end;
label 1: for i := z step 1 until K do
            begin
             if weight original [location [i]] \neq weight original [location [i+1]]
```

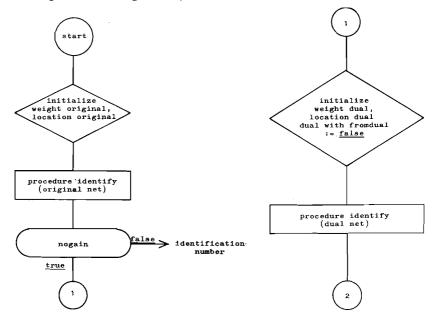
```
then go to continue
             end:
continue: if i > z
            then
             begin
              for k := z step 1 until i do weight original [location [k]]: = 0;
              label 2: min: = M * 2 \uparrow K; ready: = true;
              for l := z step 1 until i do
                begin
                 if score [location [l]] < \min \land weight original [location [l]] = 0
                 then
                  begin
                   min: = score [location [l]]; ready: = false
                  end
               end:
              if ready
              then go to continue i;
              weightstorage: = 2 * weightstorage;
              for n := z step 1 until i do
               begin
                if score [location [n]] = min
                then
                 begin
                   weight original [location [n]]: = weightstorage;
                   new location [t]: = location [n]; t: = t + 1
                 end
               end n;
              n:=i; go to label 2
             end of then of i > z
           else
            begin
              weightstorage: = 2 * weightstorage;
              weight original [location [i]]: = weightstorage;
              new location [t]: = location [i]; t: = t + 1
            end else;
continue i: z: = i + 1; if z \leq K
                         then go to label 1;
           if weightstorage \neq 2 \uparrow K
           then
            begin
             if weightstorage = maxweight
             then
```

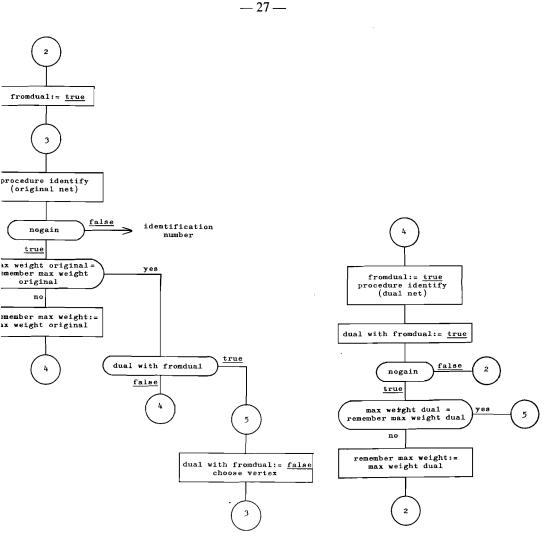
- 25 -

```
begin
    nogain: = true; go to finish
    end
    else
    begin
    for s: = 1 step 1 until K do location [s]: = new location [s];
    maxweight: = weightstorage; go to start
    end else;
    for s: = 1 step 1 until K do location [s]: = new location [s];
finish:
end identify
```

#### 3.10. Procedure identification

In the case of no gain the **procedure** identify is applied to the dual net. Then it is applied again to the original net, now calling the **procedure** with fromdual is **true**. It may happen that after coming back from the procedure there is still no gain. If the maximum weight is not increased, it is investigated whether the dual net has been used with the aid of the original net (if so, then dual with fromdual is **true**). If the maximum weight has been increased, but there is no gain, the dual net is identified (application of the **procedure** identify) with fromdual is **true**. When no further improvement can be made due to the symmetry, one of the vertices having equal weights (one with the maximum possible weight) is chosen; its weight is increased by unity. Then the **procedure** identify is called again for the original net, with fromdual is **false**. The number of choices





may be more than one; it determines the degree of symmetry of the net. If nogain is **false** the identification number corresponding to the permutation location [i] of the vertices of the original net is calculated.

The calculation of this identification number is shown in the accompanying flow chart. The associated programme is given by the **procedure** identification.

```
procedure identification (V, identificationnumber);
```

integer identificationnumber;

```
integer array V;
```

begin integer i, j, workstorage, maxweight original, maxweight dual, remember maxweight original, remember maxweight dual, n original, n dual;
Boolean dual with fromdual, fromdual, nogain, l;

integer array weight original, location original [1:K+1], inverse location [1:K], weight dual, location dual [1:M+1];

```
number of choices: = 0;
```

for i := 1 step 1 until K do

begin

weight original [i]: = 2; location original [i]: = i

location original [K+1]: = K + 1; weight original [K+1]: = 0; from dual: = false; maxweight original: = 2;

identify (K, weight original, weight dual, location original, maxweight original, n original, branch 1, branch 2, branchdual 1, branchdual 2); remember maxweight original: = maxweight original;

if ¬nogain

then go to form identificationnumber;

for i: = 1 step 1 until M do

begin

weight dual [i]: = 2; location dual [i]: = i

end;

location dual [M+1]: = M + 1; weight dual [M + 1]:=0; fromdual := false; maxweight dual: = 2; dual with fromdual: = false;

identify (M, weight dual, weight original, location dual, maxweight dual, n dual, branchdual 1, branchdual 2, branch 1, branch 2);

remember maxweight dual: = maxweight dual;

two: fromdual: = true;

three: identify (K, weight original, weight dual, location original, maxweight original, n original, branch 1, branch 2, branchdual 1, branchdual 2); if  $\neg$  nogain

then go to form identificationnumber;

if maxweight original = remember maxweight original

then

begin

if dual with fromdual

then

begin

five: weight original [location [n original]]: =

weight original [location [n original]] + 1;

number of choices: = number of choices + 1; dual with from dual: = false; go to three

#### end; go to four

end;

remember maxweight original: = maxweight original; four: fromdual: = **true**:

```
identify (M, weight dual, weight original, location dual, maxweight dual,
              n dual, branchdual 1, branchdual 2, branch 1, branch 2);
      dual with from dual: = true:
      if ¬ nogain
      then go to two;
      if maxweight dual = remember maxweight dual
      then go to five:
      remember maxweight dual: = maxweight dual; go to two;
form identificationnumber:
      for i := 1 step 1 until K do inverse location [location original [i]]: = i;
      identificationnumber: = 0;
      for l := 1 step 1 until B do
         begin
          i := K + 1 - inverse location [branch 1 [7]];
          j := K + 1 – inverse location [branch 2 [/]];
          if i > j
          then
               begin
                    workstorage: = i; i: = j; j: = workstorage
               end:
          identificationnumber := identificationnumber +
          2 \uparrow ((K \uparrow 2 + K + i * (i - 2 * K + 1) - 2 * i) \div 2)
         eпd
end identification
```

### 3.11. Input and output procedures

It is assumed that the **procedure** identification calculates an invariant of the net. Let s again be an element of  $S_N$  and s' its dual. Then the set  $S_N^*$  is built up as follows: If the number K of vertices of s is smaller than the number M of meshes, s is put in  $S_N^*$ . If K > M then s' is put in  $S_N^*$ . If K = M then of the nets s and s' that with the smaller identification number is put in  $S_N^*$ . If the net is selfdual s is put in  $S_N^*$ .

Each element of  $S_N^*$  is represented by one of its possible codes. This code is called a representative of the element of  $S_N^*$ . The set of representatives of all elements of  $S_N^*$  form the set  $T_N^*$ . Now taking one element of  $T_N^*$ , new nets are generated with the aid of the **procedure** generate nets. As soon as a new net is generated, the **procedure** form TNSTAR is called. In this procedure the identification number is calculated using the **procedure** identify, while with the aid of the **procedure** new net test it is determined whether this element of  $T_{N+1}^*$  was already found. The parameter H, which is **integer**, denotes the number of new codes of  $T_{N+1}^*$  found so far.

In the procedure form TNSTAR the procedure WRITE is used which is

described below. It writes on magnetic tape the code of a new element of  $T_{N+1}^*$ , the number of choices, an indication whether the net is selfdual or not, and an indication whether the element is the last element of  $T_{N+1}^*$  or not. In this procedure it is assumed that two new standard functions are added to the ALGOL-60 language. The first one is the **procedure** write (E), where E is an expression. This procedure writes an **integer** or **real** on magnetic tape. The second procedure is the parameterless **procedure** read, which reads the next number from magnetic tape. The format on tape determines whether the result is **integer** or **real**.

The procedures form TNSTAR, WRITE and new net test are given below.

procedure WRITE (W, number of choices, selfdual); integer number of choices;

Boolean selfdual; integer array W; begin integer i; write (W[1]), for i: = 2 step 1 until i do begin write (W[i]); if  $W[i-1] = 0 \land W[i] = 0$ then go to end end; end: write (number of choices); if selfdual

then write (1) else write (0) end WRITE;

procedure new net test (V, storage); integer storage;

```
integer array V;
```

**begin integer** *p*;

own integer array id number  $[1:4\uparrow(B-9)];$ 

for p := 1 step 1 until H do

begin

if storage = id number [p]then go to end

end;

H:=H+1; id number [H]:= storage; WRITE (V, number of choices, selfdual); end:

end new net test;

procedure form TNSTAR;

begin integer array  $U [1:2*(2*B \div 3+B)+2)];$ 

form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M); if K = M

### then

else

### begin

```
identification (V, identificationnumber);
  storage: = identificationnumber:
  dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, U);
  form branches (U, branch 1, branch 2, branchdual 1, branchdual 2,
  K, M:
  identification (U, identificationnumber);
  if identificationnumber < storage
  then
      begin
          selfdual: = false; new net test (U, identificationnumber)
       end
  else
      begin
        if identificationnumber > storage
         then
           begin
             selfdual: = false; new net test (V, storage)
           end
        else
           begin
             selfdual: = true; new net test (V, storage)
           end
      end
end
begin
  if K > M
  then
      begin
        dualize (branch 1, branch 2, branchdual 1, branchdual 2,
        K, U;
        form branches (U, branch 1, branch 2, branchdual 1, branch
        dual 2, K, M;
        identification (U, identificationnumber);
        selfdual: = false;
        new test net (U, identificationnumber)
      end
  else
        begin
```

identification (V, identificationnumber); selfdual: = false; new net test (V, identificationnumber)

#### end

#### end

end form TNSTAR

With the aid of procedure READ the code of a net, the number of choices, an indication whether the net is selfdual or not, and an indication whether the net is the last net of  $T_N^*$  or not, are read from magnetic tape. The programme is given below.

```
procedure READ (W, number of choices, selfdual, end of file);
          integer end of file, number of choices;
          Boolean selfdual;
          integer array W;
begin integer i, j;
      W[1]: = read;
     for i := 1 step 2 until i do
         begin
             W[i+1]: = read; W[i+2]: = read;
            if W[i+1] = 0 \land W[i+2] = 0
            then go to end
         end;
end: number of choices: = read;
     j: = read; if j = 0
                then selfdual: = false
                else selfdual: = true:
      end of file: = read
end READ
```

### 3.12. Complete generation and identification programme

We start with the set  $S_8^*$  consisting of one element. This element is generated by the **procedure** wheel (8). Its code is written on magnetic tape. From the set  $S_8^*$  the set  $S_9^*$  is formed, and so on. Finally the complete programme is given in programme I. It is assumed that a **procedure** stop is added to the ALGOL language. This procedure stops the machine.

# **CHAPTER 4**

### DETERMINATION OF NETWORK CURRENTS

### 4.1. Introduction

In chap. 2 it was mentioned that the rectangle dissections can be obtained from the branch currents of a net after placing an electromotive force equal to the complexity in one of the branches of the net. In a net having N branches an electromotive force can be placed in N different ways, which will lead to N dissections (possibly all different). The currents in the branches follow uniquely from Kirchhoff's laws:

- (1) The sum of the currents at any vertex is zero.
- (2) In each electrical mesh, the sum of the electromotive forces is equal to  $\Sigma I_s R_s$ , where  $I_s$  and  $R_s$  denote the branch currents and the branch resistances respectively in the mesh under consideration.

### 4.2. The branch-mesh incidence matrix

It is clear that there are M-1 independent electrical meshes of the net. For these electrical meshes a choice will be made from the M meshes of the net. Apparently there are M possible choices. In an electrical mesh a current i[m], m = 1(1)M - 1, will be assumed. The positive direction of a mesh current is that of the positive sense of the mesh. The branch currents and mesh currents are connected by the relation  $I = \Gamma i$ . Here I is the vector of the branch currents having the elements I[k], k = 1(1)B, while i is the vector of the mesh currents having the elements i[m], m = 1(1)M - 1, and  $\Gamma$  is the branch-mesh incidence matrix having B rows and M-1 columns. Furthermore we consider the vector E with elements E[k], k = 1(1)B, denoting the electromotive force in branch 1[k], branch 2[k] and the vector e with elements e[m], m = 1(1)M-1, denoting the sum of the electromotive forces in mesh m. The vectors E and e are connected <sup>12</sup>)<sup>13</sup>) by the relation  $e = \Gamma' E$ , where  $\Gamma'$  denotes the transpose of  $\Gamma$ . Now writing Z for  $\Gamma'\Gamma$ , it can be shown <sup>12</sup>)<sup>13</sup>) that e = Zi and  $I = \Gamma Z^{-1} \Gamma' E$ , where  $Z^{-1}$  means the inverse of Z. The matrix Z has M-1 rows and columns. Furthermore, Z is symmetric and non-singular.

From the definition it follows immediately that  $Z' = (\Gamma' \Gamma)' = \Gamma' \Gamma = Z$ . Hence Z is symmetric. That the matrix is non-singular follows from the fact that the branch currents are determined uniquely by the electromotive forces and the resistances in the branches of the net and from the fact that the set of mesh currents i[m], m = 1(1)M-1, is a maximal set of linearly independent mesh currents.

Now another matrix which is denoted by  $\gamma$  will be considered. It is obtained as follows: Consider the mesh currents in the *M* meshes of the net and let these currents form a vector *j*. Hence *j* has the elements j = j[m], m = 1(1)M; one

of these elements is linearly dependent on the other elements. The matrix  $\gamma$  is defined by  $I = \gamma j$ . Apparently  $\Gamma$  can be obtained from  $\gamma$  by omitting a suitable column in  $\gamma$ . In fact M different  $\Gamma$ 's can be obtained from  $\gamma$ . Since only planar networks will be considered, it is easy to see that in each row of the matrix  $\gamma$  two and only two elements are different from zero: in fact in a planar network each branch occurs in exactly two meshes. The sum of these elements is zero. The number of non-zero elements in a column is equal to the number of branches in the mesh corresponding to that column.

Next the matrix  $\zeta = \gamma' \gamma$  is formed. The matrix Z follows from  $\zeta$  by omitting one row and the corresponding column. Obviously the matrix  $\zeta$  is singular. The elements  $\zeta[r, s]$  of  $\zeta$  are either zero or minus one for  $r \neq s$ . This element is obtained by multiplying the *r*th column of  $\gamma$  by the *s*th column of  $\gamma$ . Now *r* and *s* are denoting meshes. If *r* and *s* have no branch in common this product is zero. However if *r* and *s* are incident this product equals minus one. The meshes *r* and *s* can only have one branch in common, and the positive directions of the mesh currents is such that the mesh currents in the common branches are opposite. The elements  $\zeta[i, i]$  are equal to the number of branches in mesh *i*. Hence it is clear that  $\zeta$  is the vertex-vertex incidence matrix of the dual net.

It was shown by Brooks, Smith, Stone and Tutte<sup>1</sup>) that the absolute value of all first cofactors of  $\zeta$  are equal to the complexity C of the net. This also implies that Z is non-singular.

### 4.3. Calculation of the currents

From the relation  $I = \Gamma Z^{-1} \Gamma' E$  one can obtain all possible dissections from the net. Any particular dissection is obtained by placing an electromotive force of value C in a particular branch of the net. In that case the vector E contains only one non-zero element, and the resulting vector I hence is one column of  $R = \Gamma Z^{-1} \Gamma'$  multiplied by the complexity. Therefore each column (or row) of R determines the elements of a rectangle.

The inverse of Z is obtained by using Gaussian elimination and backsubstitution. It is described in programme II and can be traced through the comments.

The matrix R can be obtained from  $ZINV = Z^{-1}$  using the following programme, where it is assumed that R and ZINV are declared as **integer array** variables; the bounds of the subscripts follow from R[1:B, 1:B] and ZINV [1:M, 1:M].

begin integer *i*, *r*, *s*; for *i*: = 1 step 1 until *M* do

begin

ZINV(i, M) := 0; ZINV(M, i) := 0

end

### end

When the branch currents are known the imperfection can be tested. It is described in the following programme, where it is assumed that the variable imperfection is **Boolean**.

**begin integer** *i*, *j*;

imperfection: = false; for i: = 1 step 1 until B-1 do for j: = i + 1 step 1 until B do

begin

if R[r, i] = R[r, j]then imperfection: = true

end

end imperfection

Furthermore zero currents can be counted. It is assumed in the following programme that the variable zero currents is declared as **integer**.

### **begin integer** *i*;

zero currents: = 0; for i: = 1 step 1 until B do

begin

```
if R[r, i] = 0
then zero currents: = zero currents + 1
end
```

end zero currents

Finally we describe the calculation of the reduction factor RF for row r of R[r, s]. The programme that calculates RF uses the **procedure** HCF(x, y) which determines the highest common factor of two integers x and y. The variable RF is **integer**.

```
begin integer l, hcf;
procedure HCF(x, y); integer x, y;
begin integer RN1, RN2;
RN1: = x; hcf: = y;
```

```
algorithm: RN2: = RN1 - hcf * (RN1 \div hcf);

if RN2 \neq 0

then

begin

RN1: = hcf; hcf: = RN2; go to algorithm

end;

hcf: = abs(hcf)

end HCF;

HCF(R[r, 1], R[r, r]);

for l: = 2 step 1 until B do HCF(R[r, l]), hcf);

RF: = hcf

end determination RF
```

# **CHAPTER 5**

### CONSTRUCTION OF BOUWKAMP CODES

### 5.1. Introduction

After having calculated the matrix R, it will be described in the sequel how the Bouwkamp codes of all dissections belonging to R can be obtained. The *k*th row or column of R is representing the currents in the branches of the original net after an electromotive force of value C has been placed in branch: branch 1[k], branch 2[k].

### 5.2. The vector ordered current

Of all vertices  $V_1, \ldots, V_K$  of the original net the respective left-cyclicordered adjacent branches are considered. Their currents are considered as elements of a vector "ordered current". The sequence of the elements of ordered current is as follows: The currents through the left-cyclic-ordered adjacent branches of vertex 1 are put into ordered current first; the currents of the leftcyclic-ordered adjacent branches of vertex 2 are put into ordered current next; and so on. Apparently ordered current has 2*B* elements.

After a column of R has been calculated, it is necessary to know where, in ordered current, a particular element of this column has to be stored positive, and where, again in ordered current, it has to be stored negative. This information is given by two vectors, namely, positive [k] and negative [k], k = 1(1)B. Hence the current in branch: branch 1[k], branch 2[k] is given by the element: ordered current [positive [k]], while the current in branch: branch 2[k], branch 1[k] is given by the element: ordered current [negative [k]].

If an element of ordered current is given we also want to know to which branch this current belongs. This information can be obtained from a vector: from [k], k = 1(1)2B. The current: ordered current [k] is flowing in branch: branch 1 [from [k]], branch 2 [from [k]]. The following relation holds: k =from [negative [k]] = from [positive [k]]. Finally we need to know for any vertex  $V_i$  the smallest l such that the branch belonging to ordered current [l]is an adjacent branch of  $V_i$ . Let this smallest l be  $l_i$ . The vector: address [k], k = 1(1)K, is defined by: address  $[k] = l_k$ .

In the next programme it is described how the vectors positive, negative, from, and address can be obtained assuming that the vectors branch 1, branch 2, branchdual 1 and branchdual 2 are given.

procedure left cyclic ordered adjacent vertices (branch 1, branch 2, branchdual 1, branchdual 2, positive, negative, address, from, K);

**begin integer** h, i, j, k, remember, meshsearch;

k:=1; i:=1; address [0]: = 0; address [1]: = 1;

search first branch: for j := 1 step 1 until B do begin if branch 1[i] = ithen begin remember: = meshsearch: = branchdual 1[i]; from [k]: = i; positive [j] := k;go to go on searching end; if branch 2[i] = ithen begin remember: = meshsearch: = branchdual 2[i]; from [k]: = i; negative [i]:=k;go to go on searching end end j;go on searching: k := k + 1;for h := 1 step 1 until B do begin if branch  $1[h] = i \wedge \text{branchdual } 2[h] = \text{meshsearch}$ then begin if branchdual 1[h] = remember then go to continue; from [k]: = h; positive [h] = k; meshsearch: = branchdual 1[h];go to go on searching end: if branch  $2[h] = i \wedge branchdual 1[h] = meshsearch$ then begin if branchdual 2[h] = remember then go to continue;

from [k]: = h; negative [h]: = k; meshsearch: = branchdual 2[h];

go to go on searching

end

**end** *h*;

continue:

i:=i+1; address [i]:=k; if  $i \neq K+1$ 

then go to search first branch

end left cyclic ordening adjacent vertices

## Example:

After applying the **procedure** left cyclic ordening adjacent vertices to the reference net we find:

<i>i</i> bran	ch 1[	i] branch	2[i] branchdu	ual 1[ <i>i</i> ] b	ranchdual	2[i] positive	[i] negative [i]
1	1	2	1		6	1	4
2	2	6	1		2	6	18
3	6	5	1		3	20	14
4	5	1	1		5	17	2
5	2	3	2		6	5	7
6	3	6	2		3	10	19
7	3	5	3		4	9	15
8	3	4	4		6	8	11
9	4	5	4		5	13	16
10	4	1	5		6	12	3
i	fro	m [i] 👘	address [i]	i	from [i]		
1		1	1	11	8		
2		4	4	12	10		
3		10	7	13	9		
4		1	11	14	3		
5		5	14	15	7		
6		2	18	16	9		
7		5	21	17	4		
8		8		18	2		
9		7		19	6		
10		6		20	3		

Next we calculate a vector: reduced ordered current. The elements of reduced ordered current are equal to the corresponding elements of ordered current divided by the reduction factor RF: reduced ordered current [k] = ordered current  $[k] \div RF$ . The following programme determines the vector: reduced ordered current. It should be noted that, to simplify notation, the vector: current [s] is identical with R[r, s] for fixed r and s = 1(1)B. **begin integer** i;

for i := 1 step 1 until B do

begin

end

reduced ordered current [positive [i]]: = current  $[i] \div RF$ ; reduced ordered current [negative [i]]: = -current  $[i] \div RF$ 

### Example:

Calculating the reduced ordered currents of the reference net, for r = 3, one obtains:

i	reduced ordered current [i]	i	reduced ordered current [i]
1	10	11	7
2	- 9	12	1
3	1	13	- 8
4	-10	14	-32
5	4	15	15
6	14	16	8
7	4	17	9
8	— 7	18	—14
9	15	19	-18
10	18	20	32

### 5.3. Determination of the Bouwkamp codes

Now we consider the left-cyclic-ordered adjacent branches of a vertex  $V_0$  and their currents:

$V_0V_1$	current $V_0V_1$
•	•
•	•
	•
$V_0 V_k$	current $V_0 V_k$

In these (cycle of) currents the first positive current following some negative current or other is searched (there are at least one positive and one negative current). The corresponding branch,  $V_0V_l$  say, is put in class  $C_{\text{pos}}$ . All successive branches  $V_0V_{l+1}$ ,  $V_0V_{l+2}$ , ...,  $V_0V_s$  that carry a positive current are put in class  $C_{\text{pos}}$ . The branch  $V_0V_{s+1}$ , carrying a negative current, is put in class  $C_{\text{neg}}$ , while all successive branches  $V_0V_{s+2}$ ,  $V_0V_{s+3}$ , ...,  $V_0V_t$  that carry a negative current are put in class  $C_{\text{neg}}$ . All indices of the second vertex are taken mod(k). Then the following theorem can be formulated:

All branches  $V_0V_1, \ldots, V_0V_k$  are belonging either to  $C_{\text{pos}}$  or  $C_{\text{neg}}$  if the network is planar.

Now a begin can be made with building up a Bouwkamp code of a dissection originating from a net after having placed an electromotive force in one of the branches of the net. This branch is called the accumulator branch. Starting from the accumulator branch we follow the current in the positive direction. If in the case of the reference net the third row of R is used, we find that the accumulator branch 6,5 is carrying a reduced ordered current equal to 32. Then one of the vertices  $V_a$  of the accumulator branch will be passed. In the reference net this is vertex 5. The next step is to consider the left-cyclic-ordered adjacent branches of  $V_a$ . In particular the branches of  $C_{pos}$  of  $V_a$  are considered. The sequence in

which they occur in  $C_{\text{pos}}$  is just the way in which the corresponding squares have to be drawn. Notice that the reduced currents of the left-cyclic-ordered adjacent branches of  $V_a$  are given by the elements: reduced ordered current [address  $[V_a]$ ], ..., reduced ordered current [address  $[V_a+1]-1$ ].

In the reference net one has:

branch	$C_{\mathrm{pos}}$	reduced ordered current
56		
53	53	15
54	54	8
51	51	9

The Bouwkamp code can be started with the reduced ordered currents belonging to the branches of  $C_{\text{pos}}$  of  $V_a$ . In the example a part of the Bouwkamp code is as follows: (15, 8, 9).

The next step is to find the vertex with which the process has to be continued. To this end a vector "contour" is defined. It is assumed that the vertex  $V_a$  has a level zero. After having drawn the Bouwkamp code so far, the vector contour contains the levels of the adjacent vertices of  $V_a$  belonging to branches of  $C_{\text{pos}}$  of  $V_a$ . The level of  $V_i$  equals the level of  $V_j$  plus the absolute value of the reduced current of branch  $V_jV_i$ . The adjacent vertices corresponding to the elements of contour are forming the vector "vertex contour". In the example of the reference net one has:

contour [1] = 15vertex contour [1] = 3contour [2] = 8vertex contour [2] = 4contour [3] = 9vertex contour [3] = 1

The next step is to find the minimum of contour [i]. In the case of more than one element equal to the minimum, the element with the smallest subscript is considered first. Let this element be contour [q]. In the example the minimum of contour equals 8, while the corresponding vertex, namely, vertex contour [2] equals 4.

The class  $C_{\text{pos}}$  of vertex contour [q] determines which squares can be drawn next. In the example one has:

branch	$C_{ m pos}$	reduced ordered current
43	43	7
41	41	1
45		

The Bouwkamp code can be extended with the reduced ordered currents of the branches of the class  $C_{pos}$  of vertex contour [q]. In the example one has: (15,8,9)(7,1). A right parenthesis will be added only if contour  $[q+1] \neq \text{contour } [q]$ .

Then the vectors contour and vertex contour are updated. The vector contour is determined as follows: The element contour [q] is replaced by the levels of the adjacent vertices of vertex contour [q]. The element vertex contour [q] is replaced by the just-mentioned adjacent vertices of vertex contour [q]. The example therefore gives:

contour $[1] = 15$	vertex contour $[1] = 3$
contour $[2] = 8 + 7 = 15$	vertex contour $[2] = 3$
contour $[3] = 8 + 1 = 9$	vertex contour $[3] = 1$
contour $[4] = 9$	vertex contour $[4] = 1$

The following step is the condensation of the vectors contour and vertex contour. If for any *i* contour [i] = contour [i+1] and vertex contour [i] = vertex contour [i+1], then the elements contour [i+1] and vertex contour [i+1] are omitted. The new vectors contour and vertex contour then have one element less than the old vectors. This process is repeated until no more elements can be omitted. Then the minimum of contour is searched again, and so on. The whole process may be stopped when both vectors contour and vertex contour have only one element. The element contour [1] will then be equal to contour  $[1] = (\text{complexity}-\text{accumulator current}) \div RF$  while vertex contour [1] will be the other vertex of the accumulator branch.

The example of the reference net is running through the following steps.

After condensation one has

contour [1] = 15vertex contour [1] = 3contour [2] = 9vertex contour [2] = 1

The minimum of contour is contour [2] and is equal to 9, while vertex contour [2] = 1. The left-cyclic-ordered adjacent branches of vertex 1 and their currents are:

branch	$C_{ m pos}$	reduced ordered current
12	12	10
15		
14		

The Bouwkamp code can be extended to (15,8,9)(7,1)(10). Updating contour and vertex contour gives

contour [1] = 15vertex contour [1] = 3contour [2] = 19vertex contour [2] = 2

There is no condensation necessary. The minimum of contour is contour [1] and is equal to 15; vertex contour [1] = 3. The left-cyclic-ordered adjacent branches of vertex 3 and their currents are:

— 42 ----

branch  $C_{pos}$  reduced ordered current 32 36 18 34 32 4 35 36

The Bouwkamp code can be extended to (15,8,9)(7,1)(10)(18,4). After updating and condensation one obtains

contour [1] = 33vertex contour [1] = 6contour [2] = 19vertex contour [2] = 2

The minimum of contour is now contour [2] and is equal to 19, while vertex contour [2] = 2. The left-cyclic-ordered adjacent branches of vertex 2 and their currents become:

branch	$C_{ m pos}$	reduced ordered current
21	26	14
23		
26		

The Bouwkamp code can be extended to (15,8,9)(7,1)(10)(18,4)(14). After updating one has:

contour [1] = 33vertex contour [1] = 6contour [2] = 33vertex contour [2] = 6

After condensation one obtains: contour [1] = 33 vertex contour [2] = 6

Now the process is ready.

It is clear that another Bouwkamp code of the same dissection would have been obtained if, instead of the left-cyclic direction, the right-cyclic direction was chosen. Furthermore other Bouwkamp codes are obtained by starting with the other vertex of the accumulator branch either using the left or the rightcyclic direction; it is then necessary to use  $C_{\text{neg}}$  instead of  $C_{\text{pos}}$ .

If we want to code the dissection with the restriction given in Bouwkamp's paper <sup>2</sup>), that the larger side is horizontal and that the left upper corner element should not be smaller than the three remaining corner elements, it is then sometimes necessary to consider the dual net also. This is so if the complexity is greater than twice the current through the accumulator branch in the original net.

The currents of the dual net can be obtained as follows. Assuming a current in the accumulator branch of the dual net equal to the current in the corresponding accumulator branch in the original net minus the complexity, the current in branch: branch 1[i], branch 2[i] of the original net is equal to the current in branch: branchdual 1[i], branchdual 2[i] of the dual net. Then the same procedure as described before can be used for obtaining Bouwkamp codes corresponding to the dual net.

The four corner elements can be obtained from the first and the last element of the set  $C_{\text{pos}}$  of the accumulator vertex  $V_{\alpha}$  and from the first and the last element of the set  $C_{\text{neg}}$  of the other vertex of the accumulator branch. The four corner elements are denoted by former first, next first, former second and next second, respectively. If in a Bouwkamp code corresponding to the original or dual net two consecutive elements are equal, a **Boolean** variable: trivial imperfection is assigned **true**. The complete procedure is given in programme II. In this programme it is assumed that two new procedures are added to the ALGOL language, namely stop and punch (E). Depending on the result of expression E, the **procedure** punch (E) punches the result in the next free columns of the punch card. The **procedure** stop stops the computer.

# CHAPTER 6

### SOME RESULTS

From the wheel  $S_8$  we obtained the sets  $S_9, S_{10}, \ldots, S_{19}$  using the electronic computers PASCAL and STEVIN of the Philips computing centre. The programmes were so arranged that the generated and identified nets could be written on magnetic tape, punched on cards or punched on paper tape. For orders up to and including 16, the list of identification numbers was stored in the core memory while for higher orders it was stored on the magnetic drum. In the latter case we applied the following sorting method.

The drum has a capacity of 16384 words of 42 bits. The identification number needs at least two words for orders higher than 16. We can therefore store 8192 identification numbers on the magnetic drum. Let the identification number *I* consist of the bits  $a_{45}, \ldots, a_3, a_2, a_1$ ; then four numbers are formed, namely,  $\sum_{k=1}^{13} a_k 2^{k-1}$ ,  $\sum_{k=1}^{13} a_{k+13} 2^{k-1}$ ,  $\sum_{k=1}^{13} a_{k+26} 2^{k-1}$  and  $\sum_{k=1}^{6} a_{k+39} 2^{k-1}$ . Let  $\frac{1}{2}A$ be the sum (modulo  $2^{13}$ ) of these four numbers. If locations *A* and *A*+1 of the magnetic drum contain zeros, then the identification number is new and is stored in these two locations. If the locations *A* and *A*+1 contain non-zero numbers, it is investigated whether the contents of *A* and *A*+1 is equal to *I*. If so, the net represented by *I* was already found. If not, the contents of the next two locations, namely, *A*+2 and *A*+3, are compared with *I*, and so on. If *I* is not found on the drum, *I* is stored in the first two locations containing zeros and following upon the locations *A* and *A*+1.

We found that the sets  $S_k^*$  have the following number of elements. The computing time on PASCAL is also given below:

k	number of c-nets	computing time
	except for duals	
8	1	
9	1	
10	2	
11	2	
12	8	
13	11	
14	37	
15	79	5 minutes
16	249	15 minutes
17	671	50 minutes
18	2182	2.5 hours
19	6692	7 hours

The codes and identification numbers of the nets of  $S_{15}^*$  and  $S_{16}^*$  were punched on cards. The cards were sorted with respect to the identification number and were listed on one of the available printers. In table I we give a fotographic copy of this output. The format is as follows: code of the net, number of choices, selfdual (1 means selfdual), identification number.

As soon as the nets were available, we investigated whether perfect or imperfect simple squared squares could be obtained from these nets. To that end we used programme II of the determination of Bouwkamp codes. However, the code was only punched if the reduced sides were equal. The codes were sorted according to increasing reduced sides.

The nets of orders 20 were generated. They were kept in the computer. The nets having a complexity satisfying the relation  $C = 2kA^2$ , where k and A are integers,  $A \ge 15$ , were punched on paper tape after they had been identified and had passed the **procedure** new net test. This programme took 30 hours of computing time.

The reason why we considered only complexities equal to  $2kA^2$  with  $A \ge 15$  was the following. We wanted to know whether perfect squared squares of order 19 exist. Now the largest element of a perfect squaring is greater than 18. If a perfect squaring exists then its reduced side is certainly greater than 19. Hence by taking  $A \ge 15$  we have not missed any simple perfect squaring of order 19. On the other hand we did not want too many nets as computer output so we chose A not too small. From experience of low-order squared squares we expect that no other simple imperfect squared squares of order 19 exist than those contained in table II.

From these nets the Bouwkamp codes of the squared squares were punched on cards. In table II we give a fotographic reproduction of the codes of the imperfect squared squares of orders up to and including 19. The format is as follows: C = complexity, S reduced horizontal side \* reduced vertical side, \* or blank (\* means imperfect, blank means perfect), RF reduction factor, Bouwkamp code, number of choices.

At last we give all Bouwkamp codes of a few nets (of orders 10, 20, 21 and 22) as typed by the on-line typewriter of PASCAL. We did not use the on-line printer because only 92 print wheels are available which is too few for the Bouwkamp codes. The same format is used as above. There are only two extra characters, namely, trivial imperfection and the number of zero currents. A reproduction of this output is given in table III.

### REFERENCES

- <sup>1</sup>) R. L. Brooks, C. A. B. Smith, A. H. Stone and W. T. Tutte, Duke math. J. 7, 312-340, 1940.
- <sup>2)</sup> C. J. Bouwkamp, Proc. Acad. Sci. Amst. 49, 1176-1188, 1946; 50, 58-78, 1296-1299, 1947 (= Indag. math. 8, 724-736, 1946; 9, 43-63, 622-625, 1947).
- <sup>3</sup>) M. Dehn, Math. Ann. 57, 314-332, 1903.
- 4) Z. Moroń, Przeglad Mat. Fiz. 3, 152-153, 1925.

- <sup>5</sup>) C. J. Bouwkamp, A. J. W. Duijvestijn and P. Medema, Catalogue of simple squared rectangles of orders nine through fourteen and their elements, Department of Mathematics, Technische Hogeschool, Eindhoven (Netherlands), May 1960, 50 pp.
- 6) T. H. Willcocks, Fairy Chess Rev. 7, Aug/Oct. 1948.
- 7) T. H. Willcocks, private communication, July 1961.
- <sup>8</sup>) R. C. Ellis, private communication, November 1959-October 1960; The perfectable rectangles of order 14, and List of 16-wire c-nets, as yet unpublished manuscripts.
- <sup>9</sup>) C. J. Bouwkamp, A. J. W. Duijvestijn and P. Medema, Tables relating to simple squared rectangles of orders nine through fifteen, Department of Mathematics and Mechanics, Technische Hogeschool, Eindhoven (Netherlands), August 1960, 360 pp.
- <sup>10</sup> J. W. Backus, F. L. Bauer, J. Green, C. Katz, J. McCarthy, P. Naur, A. J. Perlis, H. Rutishauser, K. Samelson, B. Vauquois, J. H. Wegstein, A. van Wijngaarden and M. Woodger, Numerische Math. 2, 106-136, 1960.
- <sup>11</sup>) W. T. Tutte, Proc. Acad. Sci. Amst. 64 A (= Indag. Math. 23), 441-455, 1961.
- <sup>12</sup>) W. Cauer, Theorie der linearen Wechselstromschaltungen, Akademie-Verlag Berlin, 2nd. ed. (1954), pp. 56-91.
- <sup>13</sup>) G. Kron, Tensor analysis of networks, John Wiley & Sons, Inc., New York (1939).

### TABLE I

Codes and identification numbers of  $S_{15}^*$  and  $S_{16}^*$ 

617601271023720734707467045640165105325021520354300	3	0	3777423
317301271023720134101521025620632603643046540514500	2	ŏ	7566764
617601271023720347304574056750465401641032130143100	í	õ	7775522
617601271023720347304574056750465404324026420216200	2	ŏ	7775530
617601271023720347304574056750465401641043240214200	2	ŏ	7777422
6178601271023720347304587405685046540643260216200	õ	ŏ	767524302
6285602182026720273207654704374045840148101234100	ĭ	ŏ	767524501
6178601271023720347304587405685046540643603216300	- i	ŏ	767544122
1821028320384304875405765081678034530523501256100	ò	ŏ	767560203
2842013821034830245201631043674047540612602576200	ŏ	ŏ	1365564302
201201002000000000000000000000000000000	v	v	1505554502
8568021862026720273207654704374045840148101234100	0	0	1473447250
1856101621026720273207654704374045840814801234100	õ	ŏ	1473465411
6186012781023720347304574056875046540164310321300	ĩ	ŏ	1476425530
6186012781023720347304587405685046540164310321300	i	ō	1522533622
4184012810234820145101621026720732703754301576100	i	ŏ	1563526700
8568021862026720273203765430458401481023420412400	ō	õ	1563564302
7187012810238203583046854067860476404534032174300	ĩ	ŏ	1573253401
7187012810238203583046854067860476404532402174200	ò	ŏ	1573464501
7187012810238203548304684067860457640175105321500	ĭ	ŏ	1577050322
7187012810238203583046854067860476401741045321400	ò	ŏ	1577051203
10101201020020030010051001000410401141045321480	v		1911031203
6178601271023720348730458405685046540164310321300	2	0	1625215552
6186012781023720347304587405685046540643260216200	ī	ŏ	1661633310
1821028320384304754048740576508167802345620612600	Ó	ō	1665113630
2842013821034830245201631043674047540612560576500	ō	õ	1665407350
6178601271023720348730458405685046540643260216200	ō	ō	1665413330
1821028320384304754048740576508167801234510561500	ŏ	õ	1665423314
6186012781023720347304587405685046540164104321400	ō	ō	1665423341
1821028320384304754048740576508167802345205612500	ō	õ	1665423611
2872023482045840567850165101531032130354306127600	Ó	õ	1665425511
4183401281023820145101621032673037543057650615600	ō	ō	1666130370
1821028320384304754048740576508167803456306123600	0	0	1666131134
6178601271023872034830458405685046540164310321300	0	0	1666431131
8568061860162102672027320376543045840814801234100	0	0	1667030351
1821028320384304754048740576508167803453056123500	0	0	1667031115
4184083480128102382014510162103267303754301576100	1	0	1667031213
8568021862026720273203765430458401481012310341300	1	0	1706222656
1856101621026720273207657054375045840814801234100	1	0	1706224556
8568021862026720273207657054375045840148101234100	1	0	1706424536
4854021452025632013671074170847805865068760123100	1	0	1706431217
7128702348204584067860856801651015321035430176100	0	0	1715650611
6178601271023872034830458405685046540643260216200	0	0	1725530701
4184032830813802482014510163102367420475401576100	0	0	1725550611
1821028320384304875405765081678023452025620612600	0	0	1741764302
4184013281024820145101631023672074270475401576100	1	0	1741764501
7128702348204584056785016510153103213035430176100	0	0	1745314710
8568021820862802672027320376543045840148101234100	0	0	1745315501
1821028320384304875405765081678012341014510561500	0	0	1745344522
1856101621026720273203765430458408148023420412400	0	0	1745344541
1821028320384304754048740576508167804564061234600	0	0	1745350611
2872034830823804584056785016510153210354306127600	0	0	1745360603

6178601271023872034830458405685046540432402164200	0 0	1745622701
6178601271023872034830458405685046540164104321400 1485210263202562036730741370478405865068760123100	0 0	1745660303 1746222651
1856101621026720273203765430458408148012310341300	10 00	1746224532
4184012810234820145101621026732037543057650615600	1 0	1746224551
1821028320384304754048740576501671078170123456100 1821028320384304754048740576508167801231034561300	2 <u>0</u> 0 0	1746324541 1746424531
1281087180234820458405678501651015321035430176100	õ õ	1746627003
1821028320384304754048740576508167801234104561400	0 0	1747212341
2842082380348305125024520163210367430475401576100	10	1747222603
1821028320384304754048740576508167802342045612400	0 0	1747230303
2842038230213203483024520163104367404754061257600 4184012810234204824014510162102673203754301576100	1020	1747252023 1747254013
6178601271023872034830458405685046540643603216300	2010	1747260107
6186012810238203483045840568504765402743201672100	20	1761526700
7187012810238203543048340468406786045764017532100 7187012810238203583046854067860476405325021745200	0 0 0	1765113304 1765422701
7187012810238203583068560546506786047640174532100	1 0	1766122524
7187012810238203583046854067860476401745105321500	0 0	1767012341
7187012810238203548304684067860457640175310321300	0 0	1767022505
7187012810238203583046854067860476401745310321300	0 0	1767030303
7187012810238203583046854067860476407457053217500 7187012810238203548304684067860457640753703217300	00	1767112214 1767112421
7187012810238203583046854067860476407453703217300	0 0	1767114411
7187012810238203548304684067860457407647017532100	0 0	1767142045
7187012810238203583046854067860476407453270217200 7187012810238203548304684067860457640753270217200	000	1767150015 1767150023
8218083280843808548086580876808178012341045671400	1 0	1776425003
8218083280843808548086580876808178012310345671300	1 0	1777050023
18210283203843047540487405765081678034563062360612600	0 0	767350252
18210283203343047540487405765081678023452051250561500	2 0	767524612
18210283203843047540487405765081678034530512350561500 18210283203843047540487405765081678034530523505612500	10 10	767560216 767560423
71870128102382035830468540678604764017410453403214300	0 0	777650130
71870128102382035830468540678604764017410453240218200 18210283203843047540487405765081678023452061260256200	00	777660411 1532771422
41840128102348201451016210267207327075470437401576100	2 0	1563566700
71870128102382035830468540678604764045340374303217300	1 0	1577253401
71070128102382035830468540678604764045324027420217200	0 0	1577464501
85680618601621026720273203765430458408148023420412400	1 0	1633673041
85680218208628026720273207654704374045840148101234100 28420138210348302452016310436740475406126057650625600	0 0 1 0	1663537210 1663566160
16210283203843047540467405765081678045640623460612605	100	1663566430
85680218208628026720273207657054375045840148101234100	0 0	1665361416
18210283203843047540487405765081678034530612360356300 61786012710237203487304584056850465404324026420216200	0010	1665361443 1666527430
18210283203843047540487405765081678023420612602456200	0 0	1666535130
71870128102382035830468540678604764053250245202174200 71870128102382035830468540678604764045340317430321300	0 0 0	1675631310 1675651130
1101012010250205305040054001004045540511450521500	0 0	
71870128102382035830468540678604764017410321301453100	0 0	1677424530
71870128102382035830468540678604764017410532150145100	0 0	1677425122
(18(V)28)V2382V33483V4884V8(80000000000000000000000000000000	0 0	1677431211
71870128102382035483046840678604574076470175105321500 85680218620267202732076570543750458401481012310341300	0 0 2 0	1677431211 1706424537
85680218620267202732076570543750458401481012310341300 18210283203843047540487405765081678034530562350612600	20 00	1706424537 1716430635
85680218620267202732076570543750458401481012310341300 18210283203843047540487405765081678034530562350612600 18561016210267202732076570437407547045840814801234100	20	1706424537
$\begin{array}{l} 85680218620267202732076570543750458401481012310341300\\ 182102832038430475404874405765081678034530562350612600\\ 18561016210267202732076570437407547045840814801234100\\ 13210283203843047540487405765081678012310561501345100\\ 6186012781023720347304587140586506654501641032130133100\\ \end{array}$	20 00 00 00 00 10	1706424537 1716430635 1724743462 1724745452 1724745452 1726522654
$\begin{array}{l} 85680218620267202732076570543750458w01481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 13561016210267202732076570437407547045840814801234100\\ 1321028320384304754087405765081678012310561501345100\\ 61860127810237203473045874056850465401641032130143100\\ 18210283203843047540487405765016710781702345205612500\end{array}$	2 0 0 0 0 0 0 0 1 0 1 0	1706424537 1716430635 1724743462 1724745452 1726522654 1726522662
$\begin{array}{l} 85680218620267202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18561016210267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 6186012781023720347304587405865401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501876012310456406134600\\ 1821028320384304754048740576501876012310456406134600\\ 18210283203843047540487405765018768012310456406134600\\ \end{array}$	2 0 0 0 0 0 1 0 1 0 1 0	1706424537 1716430635 1724743462 1724745452 1726522654 1726522654 1726524562
85680218520267202732076570543750458401481012310341300 18210283203843047540487405765081678034530562350612600 13561016210267202732076570437407547045840814801234100 13210283203843047540487405765081678012310561501345100 61860127810237203473045874056850465401641032130143100 18210283203843047540487405765081678012310456406134600 18210283203843047540487405765081678012310456406134600 18210283203843047540487405765016710781702345202612600 18210283203843047540487405765016710781702345620612600	2 0 0 0 0 0 1 0 1 0 1 0	1706424537 1716430635 1724743462 1724743452 1726522654 1726522652 1726522652 1726524562 1726524562
$\begin{array}{l} 85680218620267202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18561016210267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 6186012781023720347304587405865401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501876012310456406134600\\ 1821028320384304754048740576501876012310456406134600\\ 18210283203843047540487405765018768012310456406134600\\ \end{array}$	2 0 0 0 0 0 1 0 1 0 1 0	1706424537 1716430635 1724743452 1724743452 1726522654 1726522654 1726524562 1726524562 1726624552 1726630352
$\begin{array}{l} 85680218620267202732076570543750458w01481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 1856101621026720273207657043740754704584v6814801234100\\ 13210283203843047540487405765081678012310561501345100\\ 61860127810237203473045874056850465401641032130143100\\ 18210283203843047540487405765081678012310245205612500\\ 18210283203843047540487405765081678012310456406134600\\ 1821028320384304754048740576501671078170234520612400\\ 1821028320384304754048740576508167801231034530613600\\ 1821028320384304754048740576508167801231034530613600\\ 1821028320384304754048740576508167801231034530613600\\ 1821028320384304754048740576508167801231034530613600\\ 18210283203843047540487405765081678012310345305613500\\ 7187012810238203543048740546840678604576640175105321500 \\ \end{array}{0}$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743452 1724745452 1726522654 1726522652 1726524562 1726524562 1726530362 1726624552 1726630352 17266330352
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 182610267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 61860127810237203473045874058650465401641032130145100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016770781702345206134600\\ 18210283203843047540487405765016778170234520513600\\ 1821028320384304754048740576501677817023456206134600\\ 18210283203843047540487405765016778172345620613600\\ 1821028320384304754048740576501678012310345530613600\\ 18210283203843047540487405765081678012310345530513500\\ 718701281023820354304834046840678604750405324020774020\\ 71870128102382035830685056356365045047640353402774200 \end{array}$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0	1706424537 1716430635 1724743452 1724743452 1726522654 1726522654 1726524562 1726524562 1726624552 1726630352
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18261026120267202732076570437407547045840614801234100\\ 132610283203843047540487405765081678012310561501345100\\ 61860127810237203473045874058505465401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612600\\ 1821028320384304754048740576501671078170234520612600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 18210283203843047540487405765081678012310345630613600\\ 18210283203843047540487405765081678012310345630613500\\ 182102832038430475404874057650816780123103455305613500\\ 182102832038430475404874057650816780123103455305613500\\ 182102832038430475404874057650816780123103455302613500\\ 182102832038430475404874057650816780123103455302613500\\ 182102832038430475404874057650816780123103455302613500\\ 1821028320384304554068406786047640751505321500\\ 718701281023820358306856054650678604764017410532202145200\\ 718701281023820358306856054505765081474017410532502145200\\ 718701281023820358306856054505765081740174107410532502145200\\ 7187012810238203583068550545605465047640174107410532510245200\\ 71870128102382035830685505456567650476401740174105325121400\\ $ 7187012810238203583068550545505765081740174017410532510245200\\ 7187012810238203583068550545505765047640174017410532514500\\ 7187012810238203583068550545505765047640174017410532514500\\ 7187012810238203583068550545505765047640174017410532502145200\\ 71870128102382035830685505455057650476401740174105325102145200\\ 71870128102382035830685505455057650476401740174105325145200\\ 71870128102380358306855054550576504764017401741055321400\\ 7187012810238203583068550545505765047640174007410545321400\\ 71870128102380358306855054550576504764017401741055321400\\ 718701281023803583068550545505765047640174007401741055321400\\ 718701281023803583068550545505765047604774001741055321400\\ 7187012810238035830685	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743462 1724745452 1726522654 1726522652 1726524562 1726524562 1726524562 1726624552 1736630352 1736414570 1736424562 1736430651
$\begin{array}{l} 85680218620247202732076570543750458011481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 1856106210267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 61860127810237203473045874058650465401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501671078170234520612500\\ 182102832038430475404874057650167107817023456306136600\\ 1821028320384304754048740576501671078170234520512500\\ 1821028320384304754048740576501671078170234520512500\\ 7187012810238203543048340454057650167401251034532402174200\\ 718701281023820358306856054650678604764017410532502145200\\ 71870128102382035830485405786047640174105322502145200\\ 718701281023820358304856054650678604764017410532502145200\\ 718701281023820358304850546506786047640174105322502145200\\ 718701281023820358304850546506786047640174105322502145200\\ 71870128102382035830485057650615760545506780507260012500522500212600\\ 7187012810238203583048505765054650678604764017410532502145200\\ 7187012810238203583048505765054650678604764017410532502145200\\ 71870128102382035830485057505061576050678050726001250025202145200\\ 71870128102382035830485057650615760504576045045404540572402742600\\ 71870128102382035830485057505015760504576045764017410532210212600\\ 71870128102382035830485057505015760545506786047540017410532212600\\ 71870128102382035830485057505015760545506786045764017410532502145200\\ 7187012810238203583048505750501576050157605455057505015760057600576001576050057600057600576$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743652 1726522654 1726522654 1726524562 1726524562 1726524562 1726624552 1726630352 1736413650 1736414570 1736424562 1736430651 1743356250
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18261026120267202732076570437407547045840614801234100\\ 132610283203843047540487405765081678012310561501345100\\ 61860127810237203473045874058505465401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612600\\ 1821028320384304754048740576501671078170234520612600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 18210283203843047540487405765081678012310345630613600\\ 18210283203843047540487405765081678012310345630613500\\ 182102832038430475404874057650816780123103455305613500\\ 182102832038430475404874057650816780123103455305613500\\ 182102832038430475404874057650816780123103455302613500\\ 182102832038430475404874057650816780123103455302613500\\ 182102832038430475404874057650816780123103455302613500\\ 1821028320384304554068406786047640751505321500\\ 718701281023820358306856054650678604764017410532202145200\\ 718701281023820358306856054505765081474017410532502145200\\ 718701281023820358306856054505765081740174107410532502145200\\ 7187012810238203583068550545605465047640174107410532510245200\\ 71870128102382035830685505456567650476401740174105325121400\\ $ 7187012810238203583068550545505765081740174017410532510245200\\ 7187012810238203583068550545505765047640174017410532514500\\ 7187012810238203583068550545505765047640174017410532514500\\ 7187012810238203583068550545505765047640174017410532502145200\\ 71870128102382035830685505455057650476401740174105325102145200\\ 71870128102382035830685505455057650476401740174105325145200\\ 71870128102380358306855054550576504764017401741055321400\\ 7187012810238203583068550545505765047640174007410545321400\\ 71870128102380358306855054550576504764017401741055321400\\ 718701281023803583068550545505765047640174007401741055321400\\ 718701281023803583068550545505765047604774001741055321400\\ 7187012810238035830685	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743462 1724745452 1726522654 1726522652 1726524562 1726524562 1726524562 1726624552 1736630352 1736414570 1736424562 1736430651
$\begin{array}{l} 8568021862024720273207657054375045801h81012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 1856106210267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 61860127810237203473045874056850465401641032130143100\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501671078170234520612600\\ 18210283203843047540487405765016710781702345620612600\\ 1821028320384304754048740576501671078170234520612500\\ 182102832038430475404874057650816780123103455305613500\\ 7187012810238203583068560546506786047640175105321500\\ 718701281023820358306856054650678604764017410532202174200\\ 718701281023820358306856054650678604764017410532202145200\\ 718701281023820358306856054650678604764017410532212400\\ 7182102382038430475404874057650816780125305202174200\\ 718701281023820358306856054650678604764017410532202165200\\ 1821028320384304754048740576508167804764017410532202165200\\ 71870128102382035830685605465067860476401741053250212600\\ 18210283203843047540487405765081678012530526520612600\\ 182102832038430475404874057650816780125305205260212600\\ 182102832038430475404764057650815765081578045640634606123600\\ 8568061860162102672027320765705437504575045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 8568061860162102672027320765705437504575045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 85680618601621026720273207657054375045840814801234100\\ 8568061860162102672027320765705437504575045840814801234100\\ 85680618601621026720273207657054375045750458508168018201234100\\ 856806186016210267202732076570543750457504575045840814801234100\\ 85$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743652 1726522654 1726522654 1726524562 1726524562 1726524562 1726624552 1726630352 1736613050 1736414570 1736424562 1736424562 1736424562 174356250 1743566411 1746224653
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 182102832038430475404874405765081678034530562350412600\\ 18561016210267202732076570437407547045840814801234100\\ 132610283203843047540487405765081678012310561501345100\\ 618601278102372034730458714058550465401641032130145100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103455630613600\\ 182102832038430475404874057650816780123103455305613500\\ 7187012810238203583048540648067860457640475105231500\\ 718701281023820358304856054650678604764017410532502145200\\ 71870128102382035830485605465067860476401741045321400\\ 71870128102382035830485605465067860476401741045321400\\ 71870128102382035830485605465067860476401741045321400\\ 71870128102382035830485605465067860476401741045321400\\ 71870128102382035830485605465067860476401741045321400\\ 71870128102382035830485765081678035235025620612600\\ 718701281023820358304856054650678604764017410453202145200\\ 71870128102382035830465405765081678035235025620612600\\ 718701281023820358304654057650816780352350255026520612600\\ 71870128102382035830465405765081678035235025520612600\\ 718701281023820358304654057508167803545063235025520612600\\ 718701281023820358304654057650816780354506323502535025520612600\\ 71870128102382035830465405750816780354506323502535025520612500\\ 71870128102382035830465405750816780354506323502535025520612500\\ 71870128102382035830465405750816780354506323502535025520612500\\ 71870128102382035830465750816780354506323502535025520612500\\ 718701281023820358304657508167803545063235025350252500122600\\ 7187012810238203383040575081678035450632350253502525001225001\\ 71870128102382033830405750816780345500334600334606123600\\ 75508167805580816780354506334606123600\\ 75508167805580816780354506334606123600\\ 7550816780558081678035806578085806780057800578005850000\\ 75508057$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724745452 1724745452 1726522654 1726522652 1726524562 1726524562 1726624552 1726624552 1726624552 1736430550 1736414570 17364356250 17433662611
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350412600\\ 182610263203843047540487405765081678034530562350412600\\ 182610263203843047540487405765081678012310561501345100\\ 61860127810237203473045874058550467405850467401231054501345100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 18210283203843047540487405765081678012310345630613600\\ 182102832038430475404874057650816780123103455030613500\\ 71870128102382035430487405765081678012310345502612500\\ 718701281023820358306856054650678604764017410531234521500\\ 7187012810238203583068560546506786047640174105322145200\\ 7187012810238203583068560546506786047640174105322145200\\ 718701281023820358306856054650678604764017410532214500\\ 71870128102382035830685605465067860476401741023223452045200\\ 71870128102382035830685605465067860476401741053223452045200\\ 71870128102382035830685605465067860476401741053223452045200\\ 718701281023820358306856054650678604764017410532234500\\ 85680618601621026720273203765430458408148012310345300\\ 85680618601621026720273203765430458408148012310345300\\ 85680618601621026720273203765430458408148012310341300\\ 182102832038430475404814057650816780154505155015710551501\\ 1820232038430475406874057650167107817012341045541500\\ 85680618601621026720273203765430458408148012310341300\\ 1821028320384304754068740576501671078170123450454500251500\\ 85680618601621026720273203765430458018700123410056150154100\\ 18210283203843047540687405765016710781701234104551500\\ 182102832038430475406874067550016710781701234104551500\\ 1821028320384304754067560816760125410056155015700\\ 182102832038430475406755001676012541500551501500\\ 182102832038430075406750081576012541005615015700\\ 182102832038430075406874067550016710781701234104551500\\ 182102832038430075406874067550016710781701234104551500\\ 1821028320384$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724745452 1726522654 1726522654 17265224562 1726524562 1726524562 1726524562 1726624552 1736413650 1736414570 1736424562 1743666411 174356250 1746324562 1746324562 1746524114
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18261028702843047540487405765081678012310561501345100\\ 13261028203843047540487405765081678012310561501345100\\ 132612028320384304754048740576508167801231034520612310143100\\ 13210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501671078170234520512500\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576508167801231034530613600\\ 1821028320384304754048740576508167801231034530613500\\ 18210283203843047540487405765081678012310345305613500\\ 18210283203843047540487405765081678012310345302612500\\ 7187012810238203583068560546506786047640174105322102145200\\ 71870128102382035830685605465067860476401741053221400\\ 71870128102382035830685605465067860476401741053221400\\ 1821028320384304754048740576508167801350235025620612600\\ 182102832038430475404874057650816780455235025620612600\\ 182102832038430475404874057650816780457401741045321400\\ 718701281023820358306856054650678604764017410532502145200\\ 718701281023820358306856054650678604764017410532502145200\\ 718701281023820358306856054650678604764017410532502145200\\ 18210283203843047540487440576508167804550453406123600\\ 182102832038430475404874405765081678045540346061234000\\ 182102832038430475404874057650167107817012310345100\\ 1821028320384304754048740576501671078170123410561400\\ 1821028320384304754048740576501671078170123410550145100\\ 182102832038430475404874057650167107817012340345513045100\\ 18210283203843047540487405765016710781701234103541000\\ 18210283203843047540487405765016710781701234103541000\\ 18210283203843047540487405765016710781701234103541300\\ 18210283203843047540487405765016710781701234103541300\\ 18210283203843047540487405765016710781701234103541300\\ 18210283203843047540487405765016710781701234103541300\\ 1821028303843047540487540576501671078170123403541300\\ 182102830383047540487540765016710$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743452 1726522654 1726522652 1726522652 1726524562 1726524562 1726624552 1726630352 1734613650 1734613650 1736414570 1736424562 1736430651 1746324562 1746364522 1746627114
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350412600\\ 182610263203843047540487405765081678034530562350412600\\ 182610263203843047540487405765081678012310561501345100\\ 61860127810237203473045874058550467405850467401231054501345100\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 182102832038430475404874057650816780123103456306136600\\ 18210283203843047540487405765081678012310345630613600\\ 182102832038430475404874057650816780123103455030613500\\ 71870128102382035430487405765081678012310345502612500\\ 718701281023820358306856054650678604764017410531234521500\\ 7187012810238203583068560546506786047640174105322145200\\ 7187012810238203583068560546506786047640174105322145200\\ 718701281023820358306856054650678604764017410532214500\\ 71870128102382035830685605465067860476401741023223452045200\\ 71870128102382035830685605465067860476401741053223452045200\\ 71870128102382035830685605465067860476401741053223452045200\\ 718701281023820358306856054650678604764017410532234500\\ 85680618601621026720273203765430458408148012310345300\\ 85680618601621026720273203765430458408148012310345300\\ 85680618601621026720273203765430458408148012310341300\\ 182102832038430475404814057650816780154505155015710551501\\ 1820232038430475406874057650167107817012341045541500\\ 85680618601621026720273203765430458408148012310341300\\ 1821028320384304754068740576501671078170123450454500251500\\ 85680618601621026720273203765430458018700123410056150154100\\ 18210283203843047540687405765016710781701234104551500\\ 182102832038430475406874067550016710781701234104551500\\ 1821028320384304754067560816760125410056155015700\\ 182102832038430475406755001676012541500551501500\\ 182102832038430075406750081576012541005615015700\\ 182102832038430075406874067550016710781701234104551500\\ 182102832038430075406874067550016710781701234104551500\\ 1821028320384$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724745452 1726522654 1726522652 1726524562 1726524562 1726524562 1726524562 1726624552 1736430352 1736414570 1736424562 1736430651 174336650 1743666411 17466324562 1746324562 1746632111
$\begin{array}{l} 8568021862024720273207657054375045801481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 182610267202732076570437407547045840814801234100\\ 18261026720273207657043740754704584081480123100\\ 182612820384304754048740576508167801231054500\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016780123103456306134600\\ 18210283203843047540487405765016780123103456306134600\\ 1821028320384304754048740576501671078170234520612500\\ 1821028320384304754048740576501678012310345630613600\\ 182102832038430475404874057650816780123103455305613500\\ 7187012810238203583068560546506786047640175105321500\\ 718701281023820358306856054650678604764017410532502145200\\ 718701281023820358306856054650678604764017410532502145200\\ 7187012810238203843047540487405765081678045505235025620612600\\ 182102832038430475404874057650816780455840618601234100\\ 718202320384304754048740576508167804558406148012310341300\\ 182023203843047540487405765081678045584061460123410341300\\ 1821028320384304754048740576508167804558406148012310341300\\ 182102320384304754048740576508167801538406148012310341300\\ 182102320384304754048740576508167801538406148012310341300\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234104561400\\ 18210232038430475404874057650816710781701234105615045100\\ 18210232038430475404874057650816710781701234105615045100\\ 182102320384304754048740576508167107817012341056150045100\\ 1821023203843047540487405765081671078170123410561500451000\\ 182102320384304754048740576508167081570123410561500451000\\ 1821023203843047540487405765081676023420561250245200\\ 1821023203843047540487405765081678023420561250245200\\ 182102320384304754048874057650816780234205051250245200\\ 1821023203843047540487405$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724743452 1726522654 1726522652 1726524562 1726524562 1726524562 1726624552 1726624552 1736414570 1736414570 1736414570 1736430551 1746324562 1746324562 1746324552 1746324552 1746324552 1746324511 1746635111 174623511
$\begin{array}{l} 85680218620247202732076570543750458401481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18261026120267202732076570437407547045840614801234100\\ 132610283203843047540487405765081678012310561501345100\\ 6186012781023720347304587405850465401641032130143100\\ 13210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 1821028320384304754048740576501671078170234520512500\\ 1821028320384304754048740576501671078170234520512500\\ 18210283203843047540487405765081678012310345630613600\\ 18210283203843047540487405765081678012310345630613600\\ 18210283203843047540487405765081678012310345630613600\\ 182102832038430475404874057650816780123103455630613600\\ 182102832038430475404874057650816780123103455630613600\\ 182102832038430475404874057650816780123103455630613600\\ 182102832038430475404874057650816780123103455630613600\\ 18210283203843047540487405765081678012310345530613500\\ 718701281023820358306856054650678604764017410532502145200\\ 718701281023820358306856054650678604764017410532502145200\\ 718701281023820358306856054650678604764017410532502145200\\ 8568061860162102672027320376543045840814801231034506123400\\ 8568061860162102672027320376543045840814801231034506123400\\ 82680618601621026720273203765430458408148012310345061234100\\ 8268061860162102672027320376543045840814801231034506150145100\\ 182102832038430475404874505505167710781701234104551400\\ 182102832038430475404874505505167801234105615014561100\\ 18210283203843047540487405765014701781701234134561300\\ 182102832038430475404874057650167801234005615014561100\\ 182102832038430475404874057650167801234005450515015100\\ 182102832038430475404874057650167801234005450515015100\\ 18210283203843047540487405765016780123400545050152100\\ 18210283203843047540487405765081678012340054505015200\\ 1786012710238720348304584056760580167801234005450431400\\ 1821028320384304754048740576508167801234005454031400\\ 1821028320384304754048740576508167801234005454031400\\ 1821028320384304754048740576508167801234005454031400\\ 18210283203843047$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724745452 1726522654 1726522652 1726524562 1726524562 1726524562 1726524562 1726624552 1736430352 1736414570 1736424562 1736430651 174336650 1743666411 17466324562 1746324562 1746632111
$\begin{array}{l} 8568021862024720273207657054375045801481012310341300\\ 18210283203843047540487405765081678034530562350612600\\ 18261026720273207657043740754704584081480123100\\ 18261026720273207657043740754704584081480123100\\ 1826128203843047540487405765081678012310561501345100\\ 018210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016710781702345205612500\\ 18210283203843047540487405765016780123103456306134600\\ 18210283203843047540487405765016780123103456306134600\\ 1821028320384304754048740576501678012310345502613600\\ 18210283203843047540487405765081678012310345505613600\\ 18210283203843047540487405765081678012310345502613600\\ 182102832038430475404874057650816780123103455305613500\\ 7187012810238203583068560546506786047640174105322502145200\\ 718701281023820358306856054650678604764017410532202145200\\ 7187012810238203843047540487405765081678045505235025620612600\\ 182102832038430475404874057650816780455840618601234100\\ 7182023203843047540487405765081678045584061480123103445300\\ 718701281023820384304754048740576508167804558406148012310344500\\ 718701281023820384304754048740576508167804558406148012310344500\\ 718202320384304754048740576508167804558406148012310344500\\ 1821028320384304754048740576508167804558406148012310344500\\ 1821028320384304754048740576508167804558406148012310344500\\ 18210283203843047540487405765081678045584061480123103445100\\ 182102832038430475404874057650816710781701234104561400\\ 182102832038430475404874057650816710781701234104561400\\ 182102832038430475404874057650816710781701234104561400\\ 18210283203843047540487405765081671078170123410356145010\\ 182102832038430475404874057650816710781701234103561400\\ 182102832038430475404874057650816710781701234103541300\\ 182102832038430475404874057650816710781701234103551300\\ 287203483082380458405880814805765016710781701234055612045200\\ 182102832038430475404874057650816760023420561250024520216200\\ 18210223203843047540487405765081678023420561250024520216200\\ 18210223203843047540487405765081678023420561520545200\\ 18210232038430475404874057650816780$	2 0 0 0 0 0 1 0 1 0 1 0 1 0 1 0 1	1706424537 1716430635 1724745452 1726522654 1726522652 1726524562 1726524562 1726524562 1726524562 1726630352 17366130552 1736414570 1736414570 1736424562 17436564511 1746324562 1746324562 174632457114 1746627111 1746635111 1747252453

71870128102382035830685605865047840874801786710221700	1 0	1766224552
71870128102382035830685605465067860476401745310321300 71870128102382035830685605465067860476407453270217200	1 0	1766522630
18210283203843047540487405765081678023420456406124600	0 0	1766524530
71870128102382035830685605465067860476401745105321500	0 0	1766525114
71870128102382035830468540678604764074570531750321300	00	1766525122
71870128102382035430483404684067860457640753703217300		1766527104
71870128102382035430483404684067860457640175310321300	0 0	1766530330
7187012810238203543048540678604764017451053250215200	0 0	
71970129102302033030400340018004104011451033230213200	0 0	1766560431
71870128102382035830468540678604764074570532502175200	0 0	1767117401
71870128102382035430483404684067860457407647017532100	0 0	1767132304
71870128102382035830685605465067860476407457053217500	1 0	1767151106
71870128102382035830468540678604764074570517505321500	1 0	1767152124
	0 0	1767152222
71870128102382035483046840678604576407537031730321300 71870128102382035830468540678604764053250274520217200	0 0	1767153005 1767426501
718701281023820338304683408340818044184035230214320217200	0 0	
71870128102382035830468540678604764074537031730321300	0 0	1767447003 1767450131
71870128102382035830468540678604764017451032130153100	0 0	
71870128102382035830468540678604764074570217207532700	0 0	1767524302
71870128102382035483046840678604574076470753270217200 71870128102382035830468540678604764074570321730753700	1 0	1767524501
71070120102302033030400340034004804464074570321730753700	1 0	1767544122
71870128102382035830468540678604764074537021720732700	0 0	1767560203
82180832808438085480865808768081780123104564067134600	1 0	1774424570
82180832808438085480855808768081780123104567407134700	1 0	1776424530
82180832808438085480865808768081780123103456306713600	1 0	1776431211
82180832808438085480865808768081780123103453056713500	0 0	1777424302
821808328084380854808458087480817801231034353330713350	1 0	1777424501
82180832808438085480865808768081780123410456740714700	1 0	1777444122
82180832808438085480865808768081780123103456730713700	0 0	1777460203
719701289102382035483046840679860457640175101532100	1 0	77706406022
719701289102382035830468540679860647601741014532100	1 0	77706424003
189210293203943075870498540576508167802345620126100	0 0	331341252301
189210293203943075870498540576508167801234101456100	1 1	361641212322
719970199109700075+970+400+047040+774574177030		2/2//1-0002
718970128102382035483046984067960457640375301732100	0 0	363661420203
718970128102382035483046984067960457640275320172100	0 0	363665000423
719701289102382035483046840679860457640375301732100	1 0	365651404114
192102932039430758704985405765091678901234510156100	1.1	365651404122
719701289102382035830468540679860647604574017532100	0 0	365651420203
192102932039430758704985405765091678903453012356100	0 0	365655000423
719701289102382035830468540679860647603745301732100	10	371631404122
192102932039430758704985405765091678902345201256100	0 0	371631420203
719701289102382035483046840679860457404764017532100	0 0	371635000423
719701291023892035430348304698406796045764017532100		
	0 0	474257126004
812980234920469406789601765101531013210356430187100	1 0	474316526004
718970128102382035830569850546506796047640174532100	1 0 0 0	474316526004 475116152024
718970128102382035830569850546506796047640174532100 298209234590569506789601761016431046540283207138700	1 0 0 0 0 1	474316526004 475116152024 475215116420
718770128102382035830569850546506796047640174532100 298209234590569506789601761016431046540283207138700 192102932039453048654049840687609178905675012357100	1 0 0 0 0 1 0 0	474316526004 475116152024 475215116420 4752151152024
718970128102382035830569850546506796047640174532100 298209234590569506789601761016431046540283207138700 192102932039453048654049840687609178905675012357100 718970128102398203593046954067960647602453201742100	1 0 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475215152024 475217016041
718970128102382035830569850546506796047640174532100 2982092318590569506789601761016431046540283207138700 192102932039453048654049840687609178905675012357100 718970128102398203593046954067960647602453201742100 719701281023982035483046846679860457640175310132100	1 0 0 0 0 1 0 0 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475215152024 475217016041 475217032005
718970128102382035830569850546506796047640174532100 29820923459056950678960176101443104654028207138700 192102932039453048654049840687609178905675012357100 718970128102398203593046954067960647602453201742100 719701289102382035483046840679860457640175310132100 189210293203984307587048540576508167802345620126100	1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1	474316526004 475116152024 475215116420 475215152024 475215152024 475217032004 475217032005 475247026003
$718970128102382035830569850546506796047640174532100\\ 298209234590569506789601761016431046540283207138700\\ 192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701289102382035483046840679860457640175310132100\\ 1892102932039843075870485405765081678003345620126100\\ 192102932039843075870485405765091678903456301226100\\ 19210293203343075870485405765091678903456301226100\\ 19210293203945075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456507167803456507100\\ 1921029320334507587048540576509167890345650710280076579000000000000000000000000000000000$	1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217032005 475247026003 475204546041
718970128102382035830569850546506796047640174532100 298209234590569506789601761016431046540283207138700 192102932039453048654049840687609178905675012357100 718970128102398203593046954067960647602453201742100 719701281023982035483046840679860457640175310132100 189210293203984307587048540576508167802345620126100 192102932039843075870485405765081678023456301226100 192102932039843075870498540576509167803456301226100 298209234904694057896501751015321045640354307128700	1 0 0 1 0 C 0 0 0 0 0 0 0 1 0 0 0 1	474316526004 475116152024 475215116420 475215152024 475217016041 475217032005 475247026003 475304546041 47530452005
$718970128102382035830569850546506796047640174532100\\ 298209234590569506789601761016431046540283207138700\\ 192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701289102382035483046840679860457640175310132100\\ 1892102932039843075870485405765081678003345620126100\\ 192102932039843075870485405765091678903456301226100\\ 19210293203343075870485405765091678903456301226100\\ 19210293203945075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456301226100\\ 19210293203345075870485405765091678903456507167803456507100\\ 1921029320334507587048540576509167890345650710280076579000000000000000000000000000000000$	1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217032005 475247026003 475204546041
$7189701281023820358305698650546506796047640174532100\\298209234590569506789601761016431046540283207138700\\192102932039453048654049840687609178905675012357100\\718970128102398203593046954067960647602453201742100\\71970128102398203593046954067960647602453201742100\\719701281023982035483046840679860457640175310132100\\189210293203943075870485405765081678002345620126100\\19210293203943075870485405765091678903456301236100\\298209234904694057896501751015321045640354307128700\\189210293203984307587048540576508167803456301236100\\298209234904694057896501751015321045640354307128700\\189210293203984307587048540576508167803456301236100$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 47521702005 475247026003 475304546041 47530632005 475306426003
$718970128102382035830569850546506796047640174532100\\ 298209234590569506789601761016431046540283207138700\\ 192102932039453048654009840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035943046840679860457640175310132100\\ 189210293203984307587048540576508167802345620126100\\ 192102932039843075870485405765091678023456301236100\\ 19210293203984307587048540576508167803456301236100\\ 189210293203984307587048540576508167803456301236100\\ 189210293203984307587048540576508167803456301236100\\ 189210293203984307587048540576508167803456301236100\\ 219202932039843048540586506876017891012310138567100\\ 219202932039843048540586506876017891012310138567100\\ 31920293203984304854058650886506876017891012310138567100\\ 3192029320398430485405865068760178910123005805600856005800000000000000000000000$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0	474316526004 475116152024 475215116420 475215152024 475217032004 475217032005 475247026003 475304546041 475306132005 475306426003 475306426003
$718970128102382035830569850546506796047640174532100\\ 298209231890569506789601761016431046540283207138700\\ 192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035983046954067960647602453201742100\\ 719701281023982035483045840679860457640175310132100\\ 1892102932039843075870485405765081678002345620126100\\ 192102932039843075870485405765081678003456301236100\\ 2982092349046940578965017510153210456403543071287000\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540566017891012310134567100\\ 219202932039843075870485405850617891012310134567100\\ 218970128102382035830469854057650847601741014532100\\ $	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217032005 475267026003 47530632005 475306426003 475306426003 475416122424
$718970128102382035830569650546506796047640174532100\\298209234590569506789601761016431046540283207138700\\192102932039453048654049840687609178905675012357100\\718970128102398203593046954067960647602453201742100\\719701281023982035983046954067960647602453201742100\\19270293203984307587048540576508167802345620126100\\192102932039843075870485405765081678023456301236100\\298209234904694057897048540576508167803456301236100\\298209234904694057897048540576508167803456301236100\\298209234904694057895048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\2198202932039843075870485405765081678074741014522100\\3983082380934590569506789041761016432104654077200$	1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 475217032005 475247026003 475304546041 475306132005 475306426003 475416122424 475605112222 475605112222
$\begin{array}{c} 718970128102382035830569850546506476604766047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 1921029320394530486540049840687609178905675012357100\\ 718970128102398203593046954067960447602453201742100\\ 719701289102382035483046840679860457640175310132100\\ 189210293203943075870485405765091678003456301236100\\ 192102932039430758704985405765091678003456301236100\\ 2982092318904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 7189701281023820358304698540576608476017891012310134567100\\ 718970128102382035830469854067960447601741014532100\\ 71897012810238203589304698057650045764017510153210445210465407128700\\ 71897012810238203583046985405765064576017471014532100\\ 718970128102382035830459801761016432104654077128700\\ 718970128102382035893046986540649067960457640175101532100\\ 71897012810238203589304698540576509176101545210455407128700\\ 71897012810238203589304698057650045764007517407128700\\ 71897012810238203589304698540576509176107647540775101532100\\ 71897012810238203589304698057650077610761076472104554077128700\\ 718970128102358203589304698679600457640775101532100\\ 718970128102358203589304698057650075760077510754075764075764075764075764075764075764075764075764075005764075067576075760$	1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 4752217022003 475304540041 475306132005 475306426003 47530612022 475505112222 475605112222 475605113005 475605113005
$7189701281023820358305695054650546506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 7192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035483048840679860457640175310132100\\ 1892102932039430758704985405765081678002345620126100\\ 192102932039430758704985405765081678003456301236100\\ 298209234904694057896501751015321045640354507128700\\ 1892102932039843075870498540576508167803456301236100\\ 298209234904694057896501751015321045640354507128700\\ 18921023203984307587049540586506876017891012310134567100\\ 7189701281023820358304695406796047604476045761014532100\\ 398308238093459056950678960176101643210465407128700\\ 718970128102398203549304854087866456502507500175101532100\\ 7189701281023398203549304874067960457600155092013452100\\ 71897017310378303843058760485405786045565092502013452100\\$	1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475304546041 475306426003 475306426003 4754161222424 475605112222 475605113005 475606407003
$718970128102382035830569650546505760476047640174532100\\298209234590569506789601761016431046540283207138700\\192102932039453048654049840687609178905675012357100\\718970128102398203593046954067960647602453201742100\\719701281023982035483046840679860457640175310132100\\1892102932039843075870485405765081678023456301236100\\192102932039843075870485405765081678023456301236100\\298209234904694057896501751015321045640354307128700\\189210293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\718970128102382035830469854067960647601741014532100718970128102398203549304694067960457640175101532100718970128102398203543046940679604576401751015321007189701281023982035430469406796045764017510153210071897012810239820354304694067860456017501532100718970128102398203543046940678604576401751015321007189701281023982035430469406786045764015422104654071281023982035430587606457604556401550153210071897012810239820354304694065766045764055640575400755005920345200$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475215152024 475217032005 475247026003 475304586041 475306426003 475306426003 475506122422 475605112222 475605113005 475605407003 47560612208
$718970128102382035830569850546505796047640174532100\\ 298209231890569506789601761016431045540283207138700\\ 7192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960447602453201742100\\ 71970128102398203548304840679860457640175310132100\\ 189210293203943075870485405765081678002345620126100\\ 19210293203943075870485405765081678002345620126100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 18921029320398430758704854058506876017891012310134567100\\ 71897012810238203984307587048540576508167803456301236100\\ 2192029320398430758704854058506876017891012310134567100\\ 718970128102382035830495854067960457640175101532100\\ 7189701281023820354304854058560876047960457640175101532100\\ 718970128102398203549304694057604857640175101532100\\ 7189701281023982035493045876048540559509125902345200\\ 718970128102380384305876048540569509125902345200\\ 71897012810239820359204595067860456064760174014452100\\ 718970128102398203549304594065960590125920345200\\ 71897012810239820354930587604854056950590125920345200\\ 718970128102398203592045950678605901259013452100\\ 71897012810239820359205950678605876048540569509125902345200\\ 71897012810239820359205950678659674064750174014552100\\ 7189701281023982035920595065950678091740174014522100\\ 71897012810239820359205950659509125902345200\\ 7189701281023982035920595065950678091740174014522100\\ 7189701281023982035920592059500745005909125902345200\\ 7187012810239820359205950595067805909125902345200\\ 718970128102398203592059506780595067805909125902345200\\ 7187012810239820359205950595067809740055909125902345200\\ 71897012810239820359205920595067805909125902345200\\ 7189701281023982035920592059205909125902345200\\ 7189701281023982035920595059505909125902345200\\ 718970128102398203592059506595067805909125902345200\\ 7189701281023982035920595067805950675005909125902345200\\ 71897012810239820359205359067555007659509125902345200$	1 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 47521515024 475217016041 475217016041 475217022003 4752647026003 475304546041 475306132005 47530642003 475416122424 475605112222 475606112023 475606422023 475606422023 475606422023
$718970128102382035830549650544506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 718970128102398203593046954067960647602453201742100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 1892102932039843075870485405765081678023456301236100\\ 192102932039843075870485405765081678023456301236100\\ 2982092340946944057896501751015321045640354501236100\\ 29820923409469440578965017510153210456403545071236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830469854067960647601741014532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 7189701281023982035493046940679604576017410145200238200\\ 718970128102398203549304694067960457601740175101532100\\ 71897012810239820354930469406796047601740175101532100\\ 71897012810239820354930469406796047601740175101532100\\ 71897012810239820354930469406796047601740076072500345200\\ 71897012810239820354930469406796047601740076072500345200\\ 71897012810239820354930469406796076070700725002345200\\ 71897012810239820354930469540679607720072502345200\\ 71897012810239820354930469540679607720072107710778203782039430553004954057600720072107700726552200\\ 71897002802398203593046954067960677007740072076532200\\ 7189700280239820359304695406796067700774007710276532200\\ 71897002802398203593046954067960772007210276552200$	1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 4752005402003 475304546041 475306432005 475306426003 475416122424 475605112222 475605417203 475606422045 47560642203 47560642203 475607012023 575413110214
7189701281023820358305696505465057609760976097609760976097609760976097609	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 47522517032005 475304546041 475306132005 475306426003 475306122045 475506122242 475605113005 475606122045 475606122045 475606122045 475606122045 475606122045 475606122045 475606122045 475606122045 475607012023 575413110214
$718970128102382035830549650544506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 718970128102398203593046954067960647602453201742100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 1892102932039843075870485405765081678023456301236100\\ 192102932039843075870485405765081678023456301236100\\ 2982092340946944057896501751015321045640354501236100\\ 29820923409469440578965017510153210456403545071236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830469854067960647601741014532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 7189701281023982035493046940679604576017410145200238200\\ 718970128102398203549304694067960457601740175101532100\\ 71897012810239820354930469406796047601740175101532100\\ 718970128102398203549304695406796047601740175101532100\\ 71897012810239820354930469406796047601740076072500345200\\ 718970128102398203549304694067960776007200345200345200\\ 718970128102398203549304695406796077200252920345200\\ 7189701281023982035493046954067960772007200345200\\ 7189701281023982035930469540679607720072072053202305200\\ 71897012810239820359304695406796077200720720755202345200\\ 7189701281022398203593046954067960677007740077400772076532200\\ 718970072802398203593046954067960677007740077207652202345200\\ 718970728023982035930469540659506760077407072076532200\\ 718970072802398203593046954067960772007207652202345200\\ 718970072802398203593046954065950676007740772076532200\\ 7189700728022398203593046954065950676007740770276532200\\ 7189700728022398203593046954065950676007740707207653200\\ 718970072802239820393046954065950676007740707207653200\\ 7189700728002398203943045540675406776007740707207650760772076522020\\ 71897$	1 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 4752005402003 475304546041 475306432005 475306426003 475416122424 475605112222 475605417203 475606422045 47560642203 47560642203 475607012023 575413110214
$718970128102382035830569650546505760476047640174532100\\298209234590569506789601761016431046540285207138700\\192102932039453048654049840687609178905675012357100\\718970128102398203593046954067960647602453201742100\\719701281023982035483046840679860457640175310132100\\189210293203943075870485405765081678023456301236100\\19210293203943075870485405765081678023456301236100\\298209234904694057896501751015321045640354307128100\\189210293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\21920293203984307587048540576508167803456301236100\\7189701281023820358304698540679606476017410145321007189701281023982035493046940679604576401751015321007189701281023982035430458405760845405755025902134521007189701281023982035430458405860476045764017510153210071897012810239820354304584058604760457640175101532100718970128102398203543045840586047604576401751015321007189701281023982035430458405860476045764017510153210071897012810239820354304584058506595092021590234521007189701281023982035430458405850657604854056702592013452100718970128102398203593046954067960176017410145321026071897012810239820359304695405760476017410145321007189701281023982035930469540576047601741014532100718970128102398203593046954057604760174101453210071897012810239820359304695405760476017210276543200718970128102398203593046954057560676017210276543200819770017210202392035493046940679601721027654320081977001291023920354930469406796017210276543200819780129102392035493046940679605867601721027654320081978012910239203549304694067960586760172102765432008197801291023920354930469406796058676017210276543200819780129102392035493046940679605867601721027654320081978012910239203549304694067960586760172802765432008197801291023920354930469406796058676017280123457100$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 47521512024 475217032005 47530456001 475306426003 475306426003 475416122424 475605112222 475605113005 475605407003 475605407003 475606122045 475607012023 575413110214 575413110214
$71897012810238203583056950546505465057960876601760174532100\\ 298209238590569506789601761016431046540285207138700\\ 71897012810239820359304695406796045760253201742100\\ 71897012810239820359304695406796045760215310132100\\ 189701281023982035483045840679604957640175310132100\\ 189210293203943075870485405765081678002385620126100\\ 298209234904694057896501751015321045640354501236100\\ 298209234904694057896501751015321045640354501236100\\ 298209234904694057896501751015321045640354507128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540586506876017891012310134567100\\ 718970128102382035830469540679604760457640175101532100\\ 7189701281023820354930458405851604786047500457604157101532100\\ 718970128102382035493045940578604760457604157101532100\\ 718970128102382035493045940585160876048540579509125902345200\\ 718970128102398203593046954067960455690125902345200\\ 819710812802398203593045940569506780017210276543200\\ 81971081280239820359304594056950576017210276543200\\ 819710812802398203593045940569506780017210276543200\\ 81971081280239820349304594056950576017210276543200\\ 819710812802398203493045940569506780017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 819710812802398203493045940569506796017210276543200\\ 8197108128023982034930459405695067960176210226543200\\ 819710812802398203493045940569506796017621026543200\\ 8197108128023982034930459405595067960176210256543200\\ 819710812802398203493045940559506796017621026543200\\ 8197108128023982034930459405595067960176210256543200\\ 8197108128023982034930459405595067960176210256543200\\ 807108128023982034930459405595506790017621025$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475287026003 475304546041 475306132005 475306426003 475416122424 475605112222 475605112222 475605122045 475606422023 475606422023 575413110214 575413110045 575611504122
$\begin{array}{l} 718770128102382035830569650546506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 192102932039453048654049840687609178905675012357100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678023456301236100\\ 19220293203943075870485405765081678023456301236100\\ 192102932039843075870485405765081678023456301236100\\ 192102932039843075870485405765081678023456301236100\\ 19220293203984307587048540576508167803456301236100\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830469854067960847601741014532100\\ 718970128102398203583046985406796045760175101532100\\ 71897012810239820358304698576048540569509125902345200\\ 7189701281023982035930469405760457601741014532100\\ 196710173210378303843065876048540569509125902345200\\ 71897012810239820359304694057960457601741014532100\\ 19701281023982035930469405760956796017201276543200\\ 718970128102398203593046940576095690125902345200\\ 71897012810239820359304694057609647601741014532100\\ 197012810239820359304694057609647601741014532100\\ 189710812802398203493046940654956796017201276543200\\ 71897012810239820359304694055906796017201276543200\\ 81971081280239820349304694057506776017210276543200\\ 8197108128023982034930469405675608796017890123457100\\ 189710801290239239420349304594056750675067160172026543200\\ 81971081280239820349304594056950679601760172025452020\\ 819770081290239820349304594056950679601760172025452020\\ 81977008129023982034930459405695067960717601231025452020\\ 8197002230239430698049654055687508675067200172025543200\\ 81971081280239820349304594056950679601760123162345201265543200\\ 89710802392039430698049654055687508760176201225452022545202\\ 89710802390394306980496540556875097601762012345201257100\\ 8071080230394306980496540556875097600760176902345201257100\\ 80710820303943069804965405568750976079601769023455201257100\\ 80710802300394500980496540556875009778002345501265435200$	1       0         0       1         0       0         0       0         0       1         0       0         1       0         0       0         1       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 475306132005 475306426003 475416122424 475605112222 4756054122045 475606422023 475606422023 475606422023 475606422023 475606422023 575413110214 575413110214 575413110214 575413110214
$\begin{array}{c} 718970128102382035830569850546506796047640174532100\\ 298209231890565966789601745101431046540283207138700\\ 19210293203945304865404984087760647602453207138700\\ 7189701281023982035983046754047960647602453201742100\\ 71870128102398203548304840679860457640175310132100\\ 189210293203943075870485405765081678002345620126100\\ 19210293203943075870485405765081678003456301236100\\ 298209234904694057896501751015321045640354307128700\\ 7189701281023820358304698540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 7189701281023820358304698540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 71897012810238203583046985406776044760711014532100\\ 718970128102382035830469854067760478071171014532100\\ 71897012810239820358304698540677604750717101532103857100\\ 7189701281023982035830469854067560847607741014532100\\ 71897012810239820358304698540676047607141014532100\\ 7189701810239820359304695406766047601711014532100\\ 718970181280239820359304695406766047601741014532100\\ 71897018128023982035930469540676604760174014552100\\ 71897018218023982035930469540676604760174014532100\\ 7189701821802398203493045940569506796017621025453200\\ 7189701821802398203493045940569506796017621025453200\\ 189710812802398203493045940569506796017621025453200\\ 1921029320394306986049654905765078017210276543200\\ 18971081280239820349304594056950679601762102545200\\ 192102932039430698604965495675068760917890123457100\\ 18971081280239820349304594056950679601762102543200\\ 192102932039430698604965495675068760917890123457100\\ 189710812802398203493045940569506796017621025543200\\ 1921029320394306980496549550780017621025543200\\ 192102932034300584504965905796017621025543200\\ 19210293203943069860496549569506796017621025543200\\ 19210293203943069860496549569506796017621025543200\\ 1921029203943069804965495595679601762102551205\\ 102532039430698049654955950679601762102551205\\ 10571081280223982034930459405595067960176201255102584200\\ 192102920394306980496549559576867500178901762102551205\\ 10071081280239820349304594055950579601762012$	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475207026003 47530456001 475306132005 47506122045 47560512222 47560512022 475605407003 475606122045 475606122045 575413110214 575413110214 575413110214 575413110214 575413110045 575601520105 575601520203
$\begin{array}{c} 7189701281023820358305695054650546506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 718970128102398203593046954067960647602453201742100\\ 711970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678002345620126100\\ 1921029320394307587048540576508167802345620126100\\ 19210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830449854067960476017891012310134557100\\ 718970128102398203583044985406796045760457600175101532100\\ 7189701281023982035830449857640854056950292013452100\\ 71897012810239820354930469406796045760175101532100\\ 718970128102398203549304694067960457601741014532100\\ 192102932039430587640854056950590125902345200\\ 7189701281023982035930469540576085476017801741014532100\\ 19210293203943069860496540556506876001780174210276543200\\ 81978012910239820354930469406796045005690125002345200\\ 718970128102398203593046954057506876001760174210276543200\\ 81978012910239820359304695405760854056796017601741014532100\\ 7189701281023982034930459405695067960176017210276543200\\ 819780129102392035493046954056750687600176017210276543200\\ 81978012910239203549304694054956057650676017621025632200\\ 819780129102398203493045940569506796017621025632200\\ 81978012910239820349304594056950679601762122592345200\\ 81978012910239820349304594056950679601762102543200\\ 819710812802398203493045940569506796017621025543200\\ 819710812802398203493045940569506796017621025452025543200\\ 8197108128023982034930459405695067960176510225432202543200\\ 8197108128023982034930459405695067960176510225432202543200\\ 81971081280239820349304594056950679601765103554300128452102543200\\ 8197108128023982034930459405695067960176521025543200\\ 89201324123912394203469305965057605780017651031554300\\ 892103452103450304594055057807165017580721254330$		474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475304546041 475306132005 475306426003 475416122424 475605112222 475605113005 475606422023 475606422023 475606422023 475606422023 575413110214 57541311024 575413110145 575401501452
$\begin{array}{c} 718770128102382035830569650546506796047640174532100\\ 298209234590569506789601761016431046540285201138700\\ 718970128102398203593046954067960647602453201742100\\ 718970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678023456301236100\\ 19210293203943075870485405765081678023456301236100\\ 2982092349046944057896501751015321045640354501236100\\ 2982092349046944057896501751015321045640354501236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830469854067960447601741014532100\\ 71897012810239820354930469406796045760175101532100\\ 71897012810239820354930469406796045760175101532100\\ 7189701281023982035493046940679604576017410145321065407220345200\\ 718970128102398203549304694067960457601741014532100\\ 718970128102398203549304694067960457601741014532100\\ 7189701281023982035493046940554067960152502345200\\ 7189710812802398203549304694067960457601741014532100\\ 7189710812802398203549304694067960457601741014532100\\ 71897012810239820354930469406796047601741014532100\\ 718971081280239820349304694067960457601741014532100\\ 718970320394306986049654055608769017201276543200\\ 819780129023920394304694067960457601741014532100\\ 1921029320394306986049654056750679601720126553200\\ 81978012902392039430698604965405675067960176901234521025543200\\ 819710812802398203493045940569506796017621025543200\\ 89710812802398203493045940569505709707587017241025543200\\ 897108128023982034930459405695057690776017421025543200\\ 8971081280239820349304594056950679601769013521025543200\\ 89710812802398203493045940569506796017652102543200\\ 897108128023982034930459405695057690176521025543200\\ 89710812802398203493045940569505769017652102543200\\ 8982013921034930669406796059506796017658710251031564300\\ 89820139210349306594057965076907758701251031564300\\ 89710812802398203493045940559505769017658112851031564300\\ 8971081280239820349304594055950579601765871285032$	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217032005 47521512025 475247026003 475304546041 475306132005 475306426003 475416122424 475605112222 475605113005 475606122045 475606122045 475606122045 475606122045 575413110214 575413110214 575413110214 575413110214 575401501422 575601502122 575601520105 575601520105 575601520105 5756051001435
$\begin{array}{c} 7189701281023820358305695054650546506796047640174532100\\ 298209234590569506789601761016431046540285207138700\\ 718970128102398203593046954067960647602453201742100\\ 711970128102398203593046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678002345620126100\\ 1921029320394307587048540576508167802345620126100\\ 19210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035830449854067960476017891012310134557100\\ 718970128102398203583044985406796045760457600175101532100\\ 7189701281023982035830449857640854056950292013452100\\ 71897012810239820354930469406796045760175101532100\\ 718970128102398203549304694067960457601741014532100\\ 192102932039430587640854056950590125902345200\\ 7189701281023982035930469540576085476017801741014532100\\ 19210293203943069860496540556506876001780174210276543200\\ 81978012910239820354930469406796045005690125002345200\\ 718970128102398203593046954057506876001760174210276543200\\ 81978012910239820359304695405760854056796017601741014532100\\ 7189701281023982034930459405695067960176017210276543200\\ 819780129102392035493046954056750687600176017210276543200\\ 81978012910239203549304694054956057650676017621025632200\\ 819780129102398203493045940569506796017621025632200\\ 81978012910239820349304594056950679601762122592345200\\ 81978012910239820349304594056950679601762102543200\\ 819710812802398203493045940569506796017621025543200\\ 819710812802398203493045940569506796017621025452025543200\\ 8197108128023982034930459405695067960176510225432202543200\\ 8197108128023982034930459405695067960176510225432202543200\\ 81971081280239820349304594056950679601765103554300128452102543200\\ 8197108128023982034930459405695067960176521025543200\\ 89201324123912394203469305965057605780017651031554300\\ 892103452103450304594055057807165017580721254330$	1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475306426003 475306426003 475306426003 475406122424 475605112222 475605122045 475606422023 475606422023 575413110214 575413110215 575416101045 575501504122 575601504122 5756051004155 5756051004155
$\begin{array}{l} 7187701281023820358305696505465067960476017452100\\ 298209234590569506789601761016431046540245207138700\\ 718970128102398203593046954067960647602453201742100\\ 7189701281023982035983046954067960647602453201742100\\ 719701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678023456301236100\\ 19210293203943075870485405765081678023456301236100\\ 2982092349046940578965175101532104546335407123700\\ 1892102932039843075870485405765081678023456301236100\\ 298209234904694057896501751015321045640354507123700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 71897012810238203583046985406796047601741014532100\\ 71897012810239820358304698540679604760175101532100\\ 7189701281023982035830469857604854056950592013452100\\ 71897012810239820359304694067960457601741014532100\\ 198710812802398203593046940679604576017201276543200\\ 7189701281023982035930469405760950796017201276543200\\ 718970128102398203593046940554067960457601741014532100\\ 1996710173210378303843065876048540569509125902345200\\ 71897012810239820359304694055405796017201276543200\\ 71897012810239820359304694055405796017201276543200\\ 71897012810239820394304940540569505796017201276543200\\ 718970128102398203943049540569506796017690123457100\\ 1921029320394306986049654056756876017401741014532100\\ 1922029320394306986049654056750876017651022543200\\ 89710812802398203493045940569506796017652102543200\\ 89710812802398203493045940559506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 292021342303943065840565506876017481033453054300\\ 189710812802398203493045940569506796017652102543200\\ 29202134230376304594054955056768070587012851031564300\\ 189710812802398203493045940569506796017652102543200\\ 192102932039430598034930459405695067960176542102543200\\ 192102932039430598034930459405695067960176542102543200\\ 192102932039430598034954056505067760176542102543200\\ 19210293203943059804965405550675607178902345201257100\\ 1921029320394305980496$	1       0         0       1         0       0         0       1         0       0         1       0         0       0         1       0         0       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 475306132005 475306426003 475416122424 475605112222 4756054122045 475606422023 475606422023 475606422023 475606422023 575413110214 575413110214 575413110214 575413110214 5756015201045 575601520105 575601520105 575601520105 575601520105 575605100145 575605100145 575605100145 575605100145
$\begin{array}{l} 7189701281023820358305698505465067960476047604760174532100\\ 298209231890565966789601761016431046540285207138700\\ 718970128102398203593046954067960447602453201742100\\ 71197012810239820359304695406796045760178905675012357100\\ 711970128102398203598304695406796016457602345620126100\\ 1892102932039843075870485405765081678002345620126100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 71897012810238203583046985406786047891012310134567100\\ 71897012810238203583046985406786047601457640175101532100\\ 718970128102382035493045985406786047601457640175101532100\\ 71897012810238203549304598540678604760145764017501532100\\ 7189701281023820354930459405850687604854056750592013452100\\ 71897012810239820354930459405850679604576401741014532100\\ 718970128102398203549304594056950679601720276532200\\ 718970128120239320359304694067760458004760741014532100\\ 7189701281202393203549304594056950679601720125702345200\\ 7189710812802398203549304594056950679601720276532200\\ 8197801280239820349304594056950679601762102553200\\ 8197801280239820349304594056950679601762102553200\\ 8197801280239820349304594056950679601762102553200\\ 8197801280239820349304594056950679601762102553200\\ 8197801280239820349304594056950679601762102553200\\ 8197801280239820349304594056950679601762102553200\\ 81971081280239820349304594056950679601762102553200\\ 81971081280239820349304594056950679601762102553200\\ 81971081280239820349304594056950679601762102553200\\ 81971081280239820349304594056950679601762102553200\\ 81971081280239820349304594056950679601762102553200\\ 819710812802398203493045940569506796017652102553200\\ 920213921039430698604965405667500178902345201257100\\ 819710812802398203493045940569505796017652102553200\\ 9202139210394306986049654056675001768903456301235100\\ 819710812802398203493045940569506796017587012851031564300\\ 81971081280239820349304594056675001768903456301235100\\ 825502148022565201357101141414749645805865067600765803165301235100\\ 81921029320398430$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 47521702005 47526702003 475304546041 475306426003 475416122424 475605112222 475605112222 475605112222 475605407003 475607012023 575413110214 575413110214 575413110214 575601504122 575601504122 575601504152 575605100423 575605100423 630345252301 630345252301
$\begin{array}{c} 718970128102382035830569505465054650679604760174532100\\ 298209234590569506789601761016431046540285207138700\\ 718970128102398203593046954067960647602453201742100\\ 711970128102398203593046954067960647602453201742100\\ 7119701281023982035483046840679860457640175310132100\\ 189210293203943075870485405765081678002345620126100\\ 1921029320394307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 718970128102382035830469540679606476017891012310134567100\\ 718970128102382035830469540679606476017891012310134567100\\ 71897012810238203583046954067960647601741014532100\\ 398308238093459056950678960176101643210465407128700\\ 71897012810233920354930459406796045760175101532100\\ 718970128102339203549304594067960457601741014532100\\ 718970128102398203549304594057960457601741014532100\\ 718970128102398203549304594057960457601741014532100\\ 718970128102398203549304594057960457601741014532100\\ 718970128102398203549304594057960457601741014532100\\ 718970128102398203549304594055650676017740175101532100\\ 718970128102398203593046954067960457601741014532100\\ 7189701281023982035930459405545067960176210276553200\\ 71897012810239820359304594055450679601760174210276543200\\ 8197708128023982034930459405595067960176210276553200\\ 81977081280239820349304594056950679601762102545200\\ 819710812802398203493045940569506796017621025453200\\ 819710812802398203493045940569506796017621025453200\\ 189710812802398203493045940569506796017621025453200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710822039430698604965405687509178902345201237100\\ 1892102932039430758704854056850916789034552012351031564300\\ 189210293203943075870485405685091678903455201235100\\ 1525021482023920374307587048540568509167890345650123300\\ 1892102393203943075870485405685509178902345502$	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       0         1       0         0       0         1       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 475306132005 475306426003 475416122424 475605112222 4756054122045 475606422023 475606422023 475606422023 475606422023 575413110214 575413110214 575413110214 575413110214 5756015201045 575601520105 575601520105 575601520105 575601520105 575605100145 575605100145 575605100145 575605100145
$\begin{array}{l} 7189701281023820358305698505465067960476047604760174532100\\ 298209231890565966789601761016431046540285207138700\\ 718970128102398203593046954067960447602453201742100\\ 71197012810239820359304695406796045760178905675012357100\\ 711970128102398203598304695406796016457602345620126100\\ 192102932039843075870485405765081678002345620126100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 718970128102382035832469854067860475017891012310134567100\\ 7189701281023820354930459854067860475041761016452100\\ 71897012810238203549304598540678604576045760417501532100\\ 7189701281023820354930459405876048540567509125902345200\\ 718970128102398203549304594056950679601720125702345200\\ 8197801280239820354930459405675068760917890123457100\\ 1897108128023982034930459405675068760917620123457100\\ 189710812802398203493045940569506796017621025453200\\ 1921029320394306986049654956750679601762102553200\\ 1921029320394306986049654956750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 19210293203943069860496549565750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056675001789023452012551200\\ 29620139210334930459405567505796017658012851031564300\\ 189710812802398203493045940569506796017587012851031564300\\ 1897108128023982034930459405667500758700176890345501235100\\ 19210293203943075870465405667501789034550712802345201253100\\ 192102932039430758704654056875017890345670175870123457010\\ 19210293203943075870465405697507960$	1 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475304546041 475306132005 475306426003 475416122424 475605112222 475605113005 475605122045 475606422023 475607012023 575413110214 575413110214 575413110214 575401500122 5756015001422 5756015001422 57560510145 57560510145 57560510145 57560510145 57560510145 57560510145
$718970128102382035830569505465054650549508450617601765175012357100\\ 298209238590569506789601761016431046540285207138700\\ 71897012810239820359304695406796064760245207138700\\ 71897012810239820359304695406796064760245201742100\\ 719701281023982035843048840679860457640175310132100\\ 189210293203943075870485405765081678002385620126100\\ 298209234904694057896501751015321045640354507128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354507128700\\ 189210293203984307587048540586506876017891012310138567100\\ 7189701281023820358304695406786047806476047807023820128100\\ 718970128102382035830469540678604780704854057650916780325620128500\\ 7189701281023820359346954067860478064760457604175101532100\\ 71897012810238203593469546087604854056950590125902345200\\ 718970128102398203593469406796045760417210276543200\\ 81971081280239820359304694067960456017210276543200\\ 819710812802398203593045940569506796017210276543200\\ 819710812802398203493045940569506796017621025532100\\ 19210293203943069860496540567506796017621025643200\\ 819710812802398203493045940569506796017621025543200\\ 819710812802398203493045940569506796017621025543200\\ 19210293203943069860496540567506796017652102543200\\ 19210293203943069860496540567506796017652102543200\\ 19210293203943069860496540567506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940569506796017652102543200\\ 189710812802398203493045940559506796017652102543200\\ 189710812802398203493045940559506796017652102543200\\ 18921029320394305984056850576509178903456301235100\\ 1892102932039843075870485405565509178903456301235100\\ 189210293203984307587048540576508167803453023562012100\\ 1892102932039843075870485405765081678034530235520122100\\ 18921029320398430758704854057550816780345302355201226100\\ 18921029320398430758704$	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       1         0       0         1       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         1       0         2       1	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217016041 475217032005 475304546041 475306426003 475306426003 475306426003 4754061222424 475605112222 475605112222 475605112220 475606422023 475606422023 475606422023 475606422023 575413110214 575413110214 575413110215 575601500423 575601500423 575605100145 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143
$ \begin{array}{l} 7187701281023820358305695054650546506796047600174532100\\ 298209234590569506789601761016431046540285201138700\\ 718970128102398203593046954067960647602453201742100\\ 7119701281023982035983046954067960647602453201742100\\ 719701281023982035483046840679860457660175310132100\\ 18921029320394307587048540576508167802345620126100\\ 19210293203943075870485405765081678023456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 7189701281023382035493046986476017891012310134557100\\ 718970128102398203549304698647660476045760457600175101532100\\ 718970128102398203549304698676608476017410145210465407128100\\ 3983082380934590569567660854669604760174101452100\\ 1989710812802398203549304698647660847601741014532100\\ 199210293203943069860496540576085405679601762102765532100\\ 91971018128023982034930459405695067960176210276543200\\ 819778012910239203549304695405675068760917690123457100\\ 189710812802398203493045940569506796017621025543200\\ 8197710812802398203493045940569506796017621025543200\\ 8197710812802398203493045940569506796017621025543200\\ 819710812802398203493045940569506796017621025543200\\ 819710812802398203493045940569506796017621025543200\\ 89710812802398203493045940569506796017621025543200\\ 89710812802398203493045940569506750917690123452102543200\\ 189710812802398203493045940569506750917630123452102543200\\ 189710812802398203493045940569506750917650123513554300\\ 897108128023982034930459405695067509176501225013554300\\ 892102932039430758704985405755016176303456301235100\\ 8921029320398430758704985405755016176303455631235100\\ 1892102932039843075870498540576508167803455301235100\\ 89210293203984307587049854057650816780345502123100\\ 8921029320398430758704985405765081678034530235620126100\\ 8921029320398430458405765081678034530235620126100\\ 89210293203984304576057650$	1       0         0       0         0       1         0       0         0       1         0       0         0       1         0       0         1       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         1       1         2       1         0       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 475217032005 475306426003 475306426003 475416122424 475605112222 4756054122045 475606422023 475606422023 475606422023 475606422023 475606422023 475606422023 475601502105 575413110214 575413110214 575413110214 575413110214 575601520105 575601520105 575601520105 575601520203 575605100145 5756051000000000000000000000000000000000
$ \begin{array}{l} 7187701281023820356305698505465067960476047604760174532100\\ 298209231890565966789601761016431046540285207138700\\ 7189701281023982035930469540679604760245207138700\\ 718970128102398203548304695406796045760178902345201742100\\ 719701281023982035483045840679604657602345620126100\\ 19210293203984307587048540576508167802345620126100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 2192029320398430758704854058506876017891012310134567100\\ 718970128102382035832469854067960457640175101532100\\ 718970128102382035493045940576045760457640175101532100\\ 718970128102398203549304594056750647601741014532100\\ 718970128102398203549304594056750647601741014522100\\ 19671017321037830384305876048540567509125002345200\\ 718971081280239820349304594056950679601720276543200\\ 81978012802398203493045940569506796017621025643200\\ 192102932039430698604965495675068760917890123457100\\ 18971081280239820349304594056950679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 19210293203943069860496540567506796017621025543200\\ 1921029320394306986049654056750679601762102553200\\ 1921029320394306986049654056750679601762102553200\\ 298201392103930369800496540567506796017680123457100\\ 1897108128023982034930459405695067960176521202543200\\ 192102932039430698604965405675091789034563012351030\\ 189710812802398203493045940569506796017680123457100\\ 189710812802398203493045940569506796017680123457100\\ 189710812802398203493045940569506796017587012851031564300\\ 1897108128023982033493045940569506796017587012851031564300\\ 1892102932039430758704854057550816780345302356201236100\\ 189210293203943075870495850575081678034530235620123100\\ 189210293203943075870496540575081678034530235620123100\\ 1892102932039430758704985405755081678034530235620123100\\ 18921029320394307587049854057550816780323502$	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       1         0       0         1       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         1       0         2       1	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217016041 475217032005 475304546041 475306426003 475306426003 475306426003 4754061222424 475605112222 475605112222 475605112220 475606422023 475606422023 475606422023 475606422023 575413110214 575413110214 575413110215 575601500423 575601500423 575605100145 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143 575605100143
$ \begin{array}{l} 7187701281023820358305695054650549650546506796047600174532100\\ 298209234590569506789601761016431046540245207138700\\ 718970128102398203593046954067960647602453201742100\\ 7119701281023982035483046840679860457660175310132100\\ 18921029320394307587048540576508167802345620126100\\ 1921029320394307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 219202932039843058704854067960457601781012310134567100\\ 718970128102382035830469854067960457601741014532100\\ 398308238093459056950678960176101643210465407128100\\ 71897012810239820354930459405796045760457601741014532100\\ 7189701281023982035930469406796045760477601741014532100\\ 718970128102398203593046954067960457601741014532100\\ 718970128102398203593046954067960457601741014532100\\ 71897012810239820359304695406796047601741014532100\\ 1987108128023982034930459405695067960176017210276543200\\ 819770812802398203493045940569506796017690123457100\\ 18971081280239820349304594056950679601762102563200\\ 919210293203943069860496540567508760176510128512100\\ 19210293203943069860496540567508760176510128513153530\\ 199710812802398203493045940569506796017652102543200\\ 9121029320394306986049654056750876017652102543200\\ 912102932039430698604965405675091789012345201257100\\ 189710812802398203493045940569506796017652102543200\\ 912029320394306986049654056750817601765012363153501\\ 80770812802398203493045940569506796017652102543200\\ 91202932039430675870485405675091789023452012351031563300\\ 1897108128023982034930459405565508167803456301236100\\ 9282039430758704854057655081678034530235620123100\\ 1892102932039430758704854057655081678034530235520126100\\ 1892102932039430758704854057655081678034530235520126100\\ 1892102932039430758704854057655081678034530235620123100\\ 189210293203943075870485405765508167803453$	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       1         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         1       1         0       0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475306426003 475306426003 475306426003 475406132005 475606132005 475606122045 475605112222 475605122045 475606422023 475606422023 475606422023 575413110214 575413110214 575413110215 575601504122 575601504122 575601504122 575605100423 630345252301 630522517300 630524507700 644545130701 660544315501
$ \begin{array}{l} 718770128102382035830569850546506796017400174532100\\ 2982092318905659667896017451016431046540285207138700\\ 71897012810239820359304695406796047602453201742100\\ 718970128102398203594304695406796047602453201742100\\ 718701281023982035943075870485405765081678002345620126100\\ 19210293203943075870485405765081678002345620126100\\ 29820923490469405786501751015321045640354307128700\\ 7189701281023820358304698540576508167803456301236100\\ 29820923490469405786501751015321045640354307128700\\ 7189701281023820358304698540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167803456301236100\\ 21920293203984307587048540576508167890172310134567100\\ 71897012810238203583046985406776017891012310134567100\\ 7189701281023820358304698540677604780711014532100\\ 71897012810239820354930469406776604750741014532100\\ 71897018102398203493045940549604760746047504017281700\\ 71897018102398203493045940549506475091729203452100\\ 996710173210378303843065876048540569506796017251025423200\\ 91967101732103783038430658760485405675068760917890123452100\\ 199710812802398203493045940569506796017621025453200\\ 192102932039430698604965405675068760917890123457100\\ 19210293203943069860496540567506876017621025632200\\ 192102932039430698604965405675068760017680123154300\\ 1992102932039430698604965405675068760017682102532200\\ 192102932039430698604965405687509178002345201225100\\ 192102932039430698604965405687509178002345201225100\\ 192102932039430698604965405687509178002345201225100\\ 19210293203943058704854055675068760017682102345200\\ 19210293203943058704854056750687600176821023452010\\ 29210293203943058704854055675068159017802345201225100\\ 189710812802398203493045940556750681678001765210253200\\ 1921029320394307587049854055650681678012310134550100\\ 18921029320394307587049854055650816780345502125100\\ 1892102932039430758704854057550816780345502125100\\ 189210293203943075870485405750816780017545012310013852100\\ 718970128102382034303549406854057650816780017545012310013456100\\ 718970128102382034830498046970485405755$	1       0         0       1         0       0         0       0         0       0         0       1         0       0         0       1         0       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 47530632005 475306426003 475306426003 475605112222 475605113005 475606122045 475606122045 475606122045 475606122045 475606122045 475606122045 475606122045 475601012023 575413110214 575413110214 575413110214 575413110214 5756015021223 575601502122 575601502122 575601502122 57560150215 57560150215 5756010423 575605100415 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 57560510045 5756050045 57500045 575050045 5750050045 5750050045 5750050045 57500045 57
7189701281023820356305695054650679604760174532100         298209234590569506789601761016431046540285201138700         718970128102398203593046954067960647602453201742100         718970128102398203593046954067960647602453201742100         7197012810239820359304695406796064760245760123510132100         189210293203943075870485405765081678002385620126100         19201292203943075870495405765081678002385620126100         298209234904694057896501751015321045640354507128700         1892102932039843075870485405765081678003456301236100         298209234904694057896501751015321045640354507128700         189210293203984307587048540586506876017891012310138567100         71897012810238203583046954067860470504876017891012310138567100         718970128102398203583046954067860476047891012310138567100         71897012810239820359304694067960457604171014532100         71897012810239820359304695406786047601740175101532100         7189701281023982035930469540679604760171210276543200         196710812802398203493045940569506796017210276543200         197801291023920354930469406796045607176017421026543200         192102932039430698049654056750679601762102551200         1921029320394306980496540567506796017621025543200         1921029320394306980496540567506796017652102543200         19210293203943069804965405675051796017652102543200         192102932039430598045940569506796017690123457100         189710812802398203	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       1         0       0         1       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         0       0         1       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0         0       0	474316526004 475116132024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475304546041 475306132005 475306426003 475416122424 475605112202 475605113005 475605120105 475606422023 475607012023 575413110214 575413110214 575413110214 575401500122 575601500125 5756015001425 5756015001425 575605100423 57560512203 575605100423 575605100453 57560500453 57560500453 57560500453 57560500455 57560500455 575605004
$ \begin{array}{l} 718770128102382035830569850546506796017400174532100\\ 2982092318905659667896017451016431046540283207138700\\ 19210293203945304865404984087960647602453201742100\\ 718970128102398203548304854067960647602453201742100\\ 7187012810239820354830484067960647602453201742100\\ 7197012810239203943075870485405765081678003456301236100\\ 29820923490469405786501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 29820923490469405786501751015321045640354307128700\\ 7189701281023820358304698540576508167803456301236100\\ 21920293203984307587048540566017891012310134557100\\ 71897012810238203583046985406786047601741014532100\\ 7189701281023820358304698540676508167803456301236100\\ 71897012810239820359304695406760047601741014522100\\ 71897012810239820359304695406796017510153210854007800\\ 7189701281023982035930469540679601741014522100\\ 19671017310378303843065876048540569509125902345200\\ 718970128102398203593046954067960647601741014532100\\ 189710812802398203493045940569506796017621024543200\\ 192102932039430698604965405675068760917890123457100\\ 192102932039430698604965405675068760917890123457100\\ 192102932039430698604965405675068760917890123452102\\ 192203203943069860496540567506876017451023452022543200\\ 1921029320394306986049654056750687600176210256322200\\ 19210293203943069860496540567506876001762102543200\\ 19210293203943069860496540567506876001768012345210253200\\ 192102932039430587048540576501876017621025543200\\ 19210293203943058704854057650178007587012851031554300\\ 18971081280239820349304594055950576001768017621025543200\\ 19210293203943075870498540576508167800176345210234520125100\\ 189710812802398203493045940559506176017587012851031554300\\ 18921029320394307587049854057550816780017587012851031554300\\ 189210293203943075870498540575508167800345301235100\\ 18921029320394307587049854057550816780345501235100\\ 1892102932039430758704985405755081678034550123501025100\\ 189210293203943075870498540575508167803453023520126100\\ 189210293203943075870498540575508167803453023520126100\\ 718970128102382033430547948540575508167$	1       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0	474316526004 475116152024 475215116420 475215152024 475217032005 475215152024 475217032005 475304546041 475217032005 475306426003 475306426003 475306426003 47560512222 47560541222045 475606422023 475606422023 475606422023 475606422023 475606422023 475606422023 475601502105 575413110214 575413110214 575413110214 57541310214 575601520105 575601520105 575601520105 575601520105 575605100145 575605100015 57560500000000000000000000000000000000
$\begin{array}{l} 7187701281023820358305698505465067960476047604760174532100\\ 298209231890565966789601761016431046540285201138700\\ 718970128102398203593046954067960447602453201742100\\ 7119701281023982035483048406796045760178905675012357100\\ 71197012810239820354830484067960457602345620126100\\ 1921029320394307587048540576508167802345620126100\\ 298209234904694057896501751015321045640354307128700\\ 18921029320394307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540586506876017891012310134567100\\ 71897012810238203583046985406786047601457640175101532100\\ 71897012810238203984307587048540576508167803456301236100\\ 2192029320398430758704854058506876017891012310134567100\\ 718970128102398203583046985406786047601457640175101532100\\ 718970128102398203549304694067960457640175101532100\\ 718970128102398203549304694067960457604175101532100\\ 71897018210239820359304695406796045760171014532100\\ 199210293203943069860496540676904576017210276543200\\ 81978012802398203493045940569506796017210276543200\\ 81978012802398203493045940569506796017621025453200\\ 1921029320394306986049654056750017890123457100\\ 189710812802398203493045940569506796017621025453200\\ 192102932039430698049654056750017890123457100\\ 189710812802398203493045940569506796017621025452022542200\\ 29820139210349304698049654056750017890123452102543200\\ 192102932039430698049654056750017890123457100\\ 189710812802398203493045940569506796017652102543200\\ 19210293203943059840756508167801765012851031564300\\ 1897108128023982034930459405695067960176542102543200\\ 298201392103943058704854057650816780176542102543200\\ 29820139210394305870485405765081678017565011765012351031564300\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 1892102932039430758704854057508167803453023520126100\\ 18921029320394307587048540575081678034530235201256$	1       0         0       1         0       0         0       1         0       0         0       1         0       0         0       1         0       0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217016041 475217032005 475247026003 475306426003 475306426003 475306426003 47560511222242 47560511222045 475606422023 475606422023 475606422023 475606422023 575413110214 57541311024 57541311024 5756015001223 5756015001425 5756015001425 5756015010415 575605100145 575605202700
$\begin{array}{l} 718770128102382035630569505465064796047600178505675012357100\\ 298209234590569506789601761016431046540245207138700\\ 71897012810239820359304695406796064760245201742100\\ 711970128102398203594304695406796064760245201742100\\ 71970128102398203594307587048540576508167802345620126100\\ 2982092349046940578965076508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 18921023203984307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354507128700\\ 18921023203984307587048540576508167803456301236100\\ 2192029320398430758704854058506876017891012310134567100\\ 718970128102338203583046954067960457640175101532100\\ 7189701281023982035493045940576085669502590125592345200\\ 718970128102398203549304594057608450569509125902345200\\ 718970128102398203593046954067960457640175101532100\\ 718970128102398203593046954067960476017210276543200\\ 1996110173210378303843065876048540569506796017210276543200\\ 199710812802398203493045940569506796017210276543200\\ 19971081280239820349304594056950679601762102563200\\ 1921029320394306986049654056750876017690123452102\\ 18971081280239820349304594056950679601762102543200\\ 19210293203943069860496540567508776017621025543200\\ 1921029320394307587048540569506796017621025543200\\ 1921029320394307587048540569506796017621025543200\\ 189710812802398203493045940569506796017621025543200\\ 189710812802398203493045940569506796017621025543200\\ 189710812802398203493045940569506796017621025543200\\ 18971082120394307587048540576508167803152710210354300\\ 1892102932039430758704854057650816780315201231013545100\\ 1892102932039430758704854057650816780315201231013545100\\ 18921029320394307587048540576508167803123101345561100\\ 18921029320394307587048540576508167803153225520126100\\ 18921029320394304758074850816760315302355201256100\\ 18921029320394304758074850816760315302355201256100\\ 18921029320394304875405765081678031532235620126100\\ 18921029320394304758074850816760917890235501236100\\ $	$\begin{array}{cccccccccccccccccccccccccccccccccccc$	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217032005 475247026003 475304546041 475306426003 475306426003 475406122424 4756051122025 475605113005 475605122023 475607012023 475607012023 475607012023 575413110214 575413110214 575415101045 5754015001220 575601500122 5756015001423 575605100423 57560522517300 63052252501 630524507700
$\begin{array}{l} 7187701281023820358305698505465067960476047604760174532100\\ 298209231890565966789601761016431046540285201138700\\ 718970128102398203593046954067960447602453201742100\\ 7119701281023982035483048406796045760178905675012357100\\ 71197012810239820354830484067960457602345620126100\\ 1921029320394307587048540576508167802345620126100\\ 298209234904694057896501751015321045640354307128700\\ 18921029320394307587048540576508167803456301236100\\ 298209234904694057896501751015321045640354307128700\\ 189210293203984307587048540576508167803456301236100\\ 21920293203984307587048540586506876017891012310134567100\\ 71897012810238203583046985406786047601457640175101532100\\ 71897012810238203984307587048540576508167803456301236100\\ 2192029320398430758704854058506876017891012310134567100\\ 718970128102398203583046985406786047601457640175101532100\\ 718970128102398203549304694067960457640175101532100\\ 718970128102398203549304694067960457604175101532100\\ 71897018210239820359304695406796045760171014532100\\ 199210293203943069860496540676904576017210276543200\\ 81978012802398203493045940569506796017210276543200\\ 81978012802398203493045940569506796017621025453200\\ 1921029320394306986049654056750017890123457100\\ 189710812802398203493045940569506796017621025453200\\ 192102932039430698049654056750017890123457100\\ 189710812802398203493045940569506796017621025452022542200\\ 29820139210349304698049654056750017890123452102543200\\ 192102932039430698049654056750017890123457100\\ 189710812802398203493045940569506796017652102543200\\ 19210293203943059840756508167801765012851031564300\\ 1897108128023982034930459405695067960176542102543200\\ 298201392103943058704854057650816780176542102543200\\ 29820139210394305870485405765081678017565011765012351031564300\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 18921029320394307587048540576508167803456301236100\\ 1892102932039430758704854057508167803453023520126100\\ 18921029320394307587048540575081678034530235201256$	1       0         0       1         0       0         0       1         0       0         0       1         0       1         0       1         0       0         0       1         0       0         0	474316526004 475116152024 475215116420 475215152024 475217016041 475217016041 475217016041 475217032005 475247026003 475306426003 475306426003 475306426003 475406132005 475606122025 475606422023 475606422023 475606422023 475606422023 475606422023 475606422023 575413110214 575413140045 575401504122 575601504122 575601504122 575605100423 575605100423 575605100423 630345252301 6305225201 630524507700 644545130701 660544315501 66054212701 660544212701

.

195671017210276820283205486503843045940914901234100	0 0	670615244043
192102983203843048754057650916789012310134510156100	1 0	670621316410
		670621346042
617896012710237203487304598405695046540264320162100	0 0	
192102983203843075970489540576509167901231013456100	0 1	670621352022
192102983203843048975405765091679012310345301356100	0 1	670621352041
		670621354012
189210293203943049875405765081678012310134510156100	0 0	
967907129701731037830384304876540569502592013452100	0 0	670621546022
192102932039843047540487405765091678902342012456100	0 0	670621564003
239820345930569506789601761032430164210465407128700	0 0	670622326042
192102983203843075870485405765091678901231013456100	0 0	670623116014
,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
712967017310378303843048765405695025920234520132100	0 0	670623116041
719701289102382035430349830469406796045764017532100	0 0	670623216012
		670624322414
189210293203943075870498540576508167802345201256100	0 0	
189710812802382034983045940569506796017654101432100	0 1	670624322441
718970128102382035498304694067960457640275320172100	0 0	670625144016
189210293203943075870498540576508167803453012356100	υo	670625144043
718970128102398203593046954067960647603453017432100	1 0	670625160007
189710812802382034983045940569506796017621026543200	0 1	670704306422
189710812802382034983045940569506796017651015432100	0 1	670704314411
192102932039453048654049840687609178902356201267100	0 0	670704606212
719701291023892035483046840679860457404764017532100	0 0	670704642015
192102932039453048654049840687609178902356720127100	. 0 0	670704644013
189210293203943075987049540576508167803456301236100	0 0	670706206016
192102932039843075870485405765091678903453012356100	0 0	670706206043
192102983203843075870485405765091678903456301236100	õõ	670706222007
719701289102382035483046984067960457404764017532100	1 1	670706406023
192102932039430758704985405765091678901231013456100	0 0	671302236003
718970128102382049840354830946906796045764017532100	0 0	671420347042
719701289102382035498304694067960457640175310132100	0 0	671422126422
189101921029320394307587049854057650816780123456100	0 0	671422126441
107101721027520374301301047654051650616160125456100	0 0	071422120441
189210293203984307587048540576508167802342012456100	0 1	671422134122
		671422226142
719702892012910823803548304684067986045764017532100	0 1	
192102932039843075870485405765091679078970123456100	10	671424107422
189210293203943075987049540576508167801234101456100	0 Ū	671600335003
617960127102389720348304598405695046540164101432100	0 0	671600364043
189210293203943075870498540576508167801234510156100	0 1	671601217003
396302159320236202674201478510589506976079870124100		
	0 0	671601246043
	0 0	671601252023
1 1 7 1 0 12 7 1 0 2 3 6 7 2 0 3 6 3 0 4 6 7 6 3 4 0 6 1 7 6 0 6 4 6 0 1 1 4 3 3 1 0 1 3 2 1 0 0		
719701291023892035830469854067960647601745310132100 7197012910238920354830469854067960657640175310132100		
719701291023892035483046984067960457640175310132100	0 0	671601254013
		671601254015
719701291023892035483046984067960457640175310132100	0 0	
719701291023892035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100	0 0 0 0	671602226016
7 19 70 129 102389 2035 4830 4698 406 7960 4576 40 1753 10 132 100 7 18970 128 10238 2035 4830 4698 406 7960 4576 40 1753 10 132 100 7 18970 128 10238 2035 830 46985 406 7960 64 760 17453 10 132 100	0 0	671602226016 671603016023
7 19 70 129 102389 2035 4830 4698 406 7960 4576 40 1753 10 132 100 7 18970 128 10238 2035 4830 4698 406 7960 4576 40 1753 10 132 100 7 18970 128 10238 2035 830 46985 406 7960 64 760 17453 10 132 100	0 0 0 0	671602226016
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 19310423940495405865505985068760917890245620132267100	0 0 0 0 0 0	671602226016 671603016023 671604206162
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 193104239404954058650598506876091789024562013267100 189210293203943075987049540576508167804564012346100	0 0 0 0 0 0 0 0 0 1	671602226016 671603016023 671604206162 671604207016
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 19310423940495405865505985068760917890245620132267100	0 0 0 0 0 0	671602226016 671603016023 671604206162
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 193104239404954058650598506876091789024562013267100 189210293203943075870495405765508167804564012346100 192102932039430758704984048540576509167890123456100	0 0 0 0 0 0 0 0 0 1 0 0	671602226016 671603016023 671604206162 671604207016 671604207016 671604207043
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 19310423940495405865059850687660917890245620132267100 18921029320394307587049540576508167804564012346100 19210293203943075870498408540576509167890123456100 719701289102382035498304694067960457640175101532100	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	671602226016 671603016023 671604206162 671604207016 671604207013 671604207043
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035830469854067960647601745310132100\\193104239404954058650598506876091789024562013267100\\189210293203943075987049540576508167809157890123456100\\1921029320394307598704984048540576509167890123456100\\7197012891023820354803485405765091678091532450100\\7197012891023782037480495987405695046540154101432100$	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1	671602226016 671603016023 671604206162 671604207016 671604207018 671604227043 671604223007
719701291023802035483046984067960457640175310132100 718970128102382035483046984067960457640175310132100 718970128102382035830469854067960647601745310132100 19310423940495405865059850687660917890245620132267100 18921029320394307587049540576508167804564012346100 19210293203943075870498408540576509167890123456100 719701289102382035498304694067960457640175101532100	0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0	671602226016 671603016023 671604206162 671604207016 671604207013 671604207043
$719701291023802035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\7189701281023820358304698540679604576401745310132100\\1931042394049540586505985068760917890245620132267100\\189210293203943075987049540576508167800123456100\\192102932039430758704984085850576509167890123456100\\71970128910238203445705987049540576509167890123456100\\71970128910238203445405406796045761164101452100\\6196012891023420344695406596048540164101432100$	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	671602226016 671603016023 671604206162 671604207016 671604227003 671604223007 671604223007 671604242063 67160424063
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035830469854067960457640175310132100\\193104239404954058650598506876001789024562013267100\\18921029320394307598704954057650816780042545013264100\\1921029320394307598704984048540576509167890123456100\\719701289102382035483045940576509167890123456100\\619601289102378203548304594057650565506164101432100\\6196012891023820359830469540576506164101432100\\719701289102382035983046954067960457601741014532100\\719701289023892035983046954067960457604175101532100$	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0	671602226016 671603016023 671604206162 671604207018 671604207018 671604223007 671604223007 671604242063 671604260017 671605066216
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\193104239404954058650598506876091789024562013267100\\1993104239404954058650598506876091789024562013267100\\1892102932039430759870495405765508167809167890123456100\\71970128910238203548075498048540576509167890123456100\\71970128910238203548054859067960457640175101532100\\619601289102378203548304698406796045764017410114532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\719701289102389203548304698405765091687003453012356100\\719701869102389203548304698405765091687003453012356100\\71970345301289102389203548304698405765091687003453012356100\\71970345301289102389203548304698405765091687003453012356100\\719703453012891023892035483046984057650048509168703453012356100\\7197034530123692035483046984057650048509168703455012365000$ 719703453012356100	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0	671602226016 671604206162 671604207016 671604207013 671604223007 671604223007 67160422063 67160422063 671605006216 671605006423
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\193104239404954058650598506876091789024562013267100\\1993104239404954058650598506876091789024562013267100\\1892102932039430759870495405765508167809167890123456100\\71970128910238203548075498048540576509167890123456100\\71970128910238203548054859067960457640175101532100\\619601289102378203548304698406796045764017410114532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\719701289102382035983046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\7197012891023892035483046984067960457640175101532100\\719701289102389203548304698405765091687003453012356100\\719701869102389203548304698405765091687003453012356100\\71970345301289102389203548304698405765091687003453012356100\\71970345301289102389203548304698405765091687003453012356100\\719703453012891023892035483046984057650048509168703453012356100\\7197034530123692035483046984057650048509168703455012365000$ 719703453012356100	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0	671602226016 671603016023 671604206162 671604207018 671604207018 671604223007 671604223007 671604242063 671604260017 671605066216
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035830469854067960457640175310132100\\193104239404954058650598506876001789024562013267100\\18921029320394307598704954057650816780042545013264100\\1921029320394307598704984048540576509167890123456100\\719701289102382035483045940576509167890123456100\\619601289102378203548304594057650565506164101432100\\6196012891023820359830469540576506164101432100\\719701289102382035983046954067960457601741014532100\\719701289023892035983046954067960457604175101532100$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0	671602226016 671604206162 671604207016 671604207013 671604223007 671604223007 67160422063 67160422063 671605006216 671605006423
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035830469854067960457640175310132100\\1931042394049540586505985068766091789024562013267100\\1892102932039430759870495405765081678045640123456100\\1921029320394307598704984048540576509167890123456100\\719701289102382035983045940576509167890123456100\\719701289102382035983045940576509167890123452100\\719701289102382035983046954067960457640175101532100\\719701289203942035983046954067960457640175101532100\\71970129102382035983046954067960457601741014532100\\71970129102382035483046984067960457601741014532100\\71970129102382035483046984067960457601741014532100\\71970129102382035483046984067960457601741014532100$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207018 671604207018 671604220708 671604223007 671604226017 671604260017 671605006216 671605006423 6716050044017
$719701291023802035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\7189701281023820354830469854067960457640175310132100\\193104239404954058650598506876091789024562013267100\\189210293203943075987049540576598167809123456100\\1921029320394307587049840485840576509167890123456100\\7197012891023820344207840045840576509167890123456100\\719701289102382034475045967440570540175101532100\\61960128910237820344750459640679604576140175101532100\\7197012891023803944954059406796045761174101452100\\719701291023803475304954067960457640175101532100\\719701291023803475304594067960457640175101532100\\7197012910238034785049740586590168700353012356100\\7197012910238074853047850467960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100\\719701291023802055830469854067960457640175101532100$	0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 0	671602226016 671604206162 671604207016 671604207043 671604207043 671604223007 671604222063 67160422063 67160422063 671605006216 67160500423 671605044017 705141155501
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035830469854067960457640175310132100\\1931042394049540586505985068766091789024562013267100\\1892102932039430759870495405765081678045640123456100\\1921029320394307598704984048540576509167890123456100\\719701289102382035983045940576509167890123456100\\719701289102382035983045940576509167890123452100\\719701289102382035983046954067960457640175101532100\\719701289203942035983046954067960457640175101532100\\71970129102382035983046954067960457601741014532100\\71970129102382035483046984067960457601741014532100\\71970129102382035483046984067960457601741014532100\\71970129102382035483046984067960457601741014532100$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207018 671604207018 671604220708 671604223007 671604226017 671604260017 671605006216 671605006423 6716050044017
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\71897012810238203583046985406796045765401745310132100\\1931042394049540586508985068766917890245562013267100\\18921029320394307598704984048540576508167804554012345100\\1921029320394307598704984048540576509167890123456100\\71970128910238203598304694067960457640175101352100\\6196012891023782035983046954067960457640175101532100\\7197012891023892035983046954067960457640175101532100\\7197012891023892035983046954067960457601741014532100\\7197012891023892035983046954067960457601741014532100\\7197012890235483046984067960457601741014532100\\7197012910238920358830469840679604570141014532100\\71970129102389203588304698540578604570141014532100\\7197012910238920358830469854057860457014701741014532100\\7197012910238920358830469854057860457014701741014532100\\7197012910238920358830469854057860457014701741014532100\\7197012910238920358830469854057860457601741014532100\\7197012910238920358830469854057860457601741014532100\\7197012910238920358830469854057860457601741014532100\\7197012910238920358830469854057860457601741014532100\\7197012910238920358830469854057860457601741014532100\\719701291023892035883046985405786059058509168703453012356100\\719701291023892035883046985405786059058509168703453012356100\\719701291023892035883046985405786059058509168703453012356100\\719701291023892035883046985405786059054765002743201672100$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 0	671602226016 671604206162 671604207018 671604207018 671604207018 6716042207043 671604223007 671604242063 671604260017 671605006216 67160500423 6716050044017 705141155501 720261526700
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 71897012810238203583046984067960457640175310132100\\ 1931042394049540586505985068760917890245620132267100\\ 1892102932039430759870495405765091678090123456100\\ 7197012891023820354983046984067960457640175101532100\\ 61960128910232639430154964406796045764011751014321100\\ 7197012891023820354983046984067960447601171110145321100\\ 71970128910238203548304698406796045764011751015321100\\ 719701293203943047850497400569501687703530123561100\\ 7197012910238920358830469854067960647600174101145321100\\ 7197012910238920358830469854067960647600174101145321100\\ 7197012910238020358830469854067960647600174101145321100\\ 9567902197202768220283205865038543045940149101234100\\ 6189601281023820354830487654056950595257240154201672100\\ 7129607173103383038483048485540569550557430134210245200\\ \end{array}$	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207013 671604223007 671604223007 6716042230017 671605006423 671605006423 671605044017 705141155501 720261526700 721060371501
$719701291023802035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\7189701281023820354830469840679604576401745310132100\\193104239404954058650598506876091789024562013267100\\1892102932039430758704954058508760917890245012346100\\192102932039430758704984048540576509167890123456100\\719701289102382025948304694048540576509167890123452100\\619601289102378203598304695406796045761174101452100\\71970128910238202558304695406796045761174101452100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\956790219720276820283205885038543045940164701234100\\618960128102382034983045940569504765402743201672100\\712970238402354830469405569504765402743201672100\\7129702384930489408540569504765402743201672100\\71297023843034830487655454059504755402743201672100\\712923203343075870485409540590157800134210245200\\7129232039430376870485405905178701532110345102$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0	671602226016 671603016023 671604206162 671604207018 671604207043 6716042207043 671604223007 671604242063 671605006216 671605006216 671605004423 671605044017 705141155501 720261526700 721060371501 7214122701
$719701291023802035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\7189701281023820354830469840679604576401745310132100\\193104239404954058650598506876091789024562013267100\\1892102932039430758704954058508760917890245012346100\\192102932039430758704984048540576509167890123456100\\719701289102382025948304694048540576509167890123452100\\619601289102378203598304695406796045761174101452100\\71970128910238202558304695406796045761174101452100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\719701291023892035483046954067960457640175101532100\\956790219720276820283205885038543045940164701234100\\618960128102382034983045940569504765402743201672100\\712970238402354830469405569504765402743201672100\\7129702384930489408540569504765402743201672100\\71297023843034830487655454059504755402743201672100\\712923203343075870485409540590157800134210245200\\7129232039430376870485405905178701532110345102$	0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207013 671604223007 671604223007 6716042230017 671605006423 671605006423 671605044017 705141155501 720261526700 721060371501
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035830469854067960457640175310132100\\ 19310423940495405865059850687660917890245620132267110\\ 1892102932039430159870495405765091678001234561100\\ 719701289102382035498304698406796047501175101532100\\ 6196012891023262354983046984067960447601175101532100\\ 719701289102382035483046954067960447601741014522100\\ 7197012891023820354830469540679606476001741014522100\\ 7197012991023303943304785040974056951637033530123551100\\ 7197012910238020354830469540679606476017410145321100\\ 7197021291023820354830469540679606476017410145321100\\ 9567902197202768202820283565336533655600149101241100\\ 6189601281023802354830499404756305550255202134210245200\\ 712702232039483075870485405765091679001310134551100\\ 712910223220394830758704854057650916796047501175101332100\\ 712910223220394830758704854057650916796047761715101332100\\ 7189701281022322039483075870485405765091679601715101532100\\ 718970128102230203548304694605765091679601715101532100\\ 71897012810223220394830758704854057659916796047601715101532100\\ 7189701281022302035483046946057659916796037640175101532100\\ 7189701281022302035483046946057659916769015750115101532100\\ 7189671028102230394830463046576599505750291331113455100\\ 7189671028102230394830575559595040575059505950595059505352505250505352505052505250505352505152505250505250505250$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0	671602226016 671604206162 671604207016 671604207013 671604227043 6716042207043 6716042242063 671604260017 671605006216 671605006423 6716050044017 705141155501 720261526700 721060371501 721240364522
$719701291023892035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\718970128102382035483046984067960457640175310132100\\193104239404954058650598506876091789024562013267100\\18921029320394307598704954057659167809123456100\\7197012891023820394307587049840858506756509167890123456100\\719701289102382039430758704984085850676509167890123456100\\7197012891023820394307587049640576509167490123162100\\619601289102378203943045940679604576114101452100\\719701291023803947504950916870604760175101532100\\719701291023803943046954067960457640175101532100\\719701291023802035883046954067960457640175101532100\\719701291023802035883046954067960457640175101532100\\7197012910238020358830469854067960457611741014532100\\71970129102380203588304698540679604576013741014532100\\71970717310378102389203588304698540679604576013741014522100\\956790219720276820283205885038543045940149101234100\\6189601281023802035883048765405695054765402743201672100712967071731037830384540876565950547654027432016721007129210232039443075870485405765091678901231013456100712921023203944307587048540576509167890123101345610071292102320394430758704854057650916780013321028220071296717310382035890469650576509174107410741074052200071296701731038420354983046940679604576407743013422000712967017310384304696540659605259207764073103321022452007129670173103842058870468965057650916778407743101345610071897012810238203588704698606766065765097640774531013456100718970128102382035887046985606976606476017410753101345010071897012810238203588704696560576509764017531013452100$	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0	671602226016 671604206162 671604207016 671604207013 6716042207013 67160422003 67160422063 67160422063 671605006423 671605006423 671605006423 6716050044017 705141155501 72060371501 72060371501 721240364522 721240364522 721240365203
$\label{eq:response} \begin{array}{c} r19701291023892035483046984067960457640175310132100\\ r18970128102382035483046984067960457640175310132100\\ r1897012810238203548304698540679604576401745310132100\\ 193104239404954058650598506876091789024552013227100\\ 189210293203943075870498404854057650816780454012345100\\ r197012891023820354830459407960457640175101532100\\ 6196012891023782034730459870498408540576509167890123455100\\ r1970128910238203548304659407960457640175101552100\\ r197012910238203548304659406796045760174101452100\\ r19701291023820354830465806796045760174101452100\\ r19701291023892035483046984067960457601741014532100\\ r19701291023892035483046984067960457601741014532100\\ r19701291023892035830469854067960457601741014532100\\ r19701291023892035830469854059168703453012356100\\ r1970129102382035830469854067960457601741014532100\\ r1970129102382035830469854067960457601741014532100\\ r1970129102382035830469854067960457601741014532100\\ r19701281023820358304594554057507455401231013456100\\ r19201282032830358304874554505950274320134210245200\\ r197012810238203548304594059015675040175101532100\\ r1970128102382035483045940590575650157640175101532100\\ r192002320394307580704854067960457640175101532100\\ r19200232039430758070485405950575050157640175101532100\\ r192002320394307580704854055050501565001532102345200\\ r1970128102382035483048745550550501576023130132455100\\ r18970128102382035498304894067960457640175101532100\\ r189701280303383075807048940679504556015780027432015123013455100\\ r189701280303383049849406795005850615780027432013101345100\\ r1897012803233430498408940679500580157800274301310132500\\ r1897012803035830469840679500580157800254001755101532100\\ r189701280303383304898048940578058015780015780013501321020\\ r18970128030338304874057500580157800254500155015501052010321002\\ r189701283003300787078098740575058015780025450015501052050018850037550103020510027455010238500355000000000000000000000000000000$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207018 671604242063 671604242063 671605004017 671605006423 6716050044017 705141155501 720261526700 721060371501 721240364522 721240565103
$\label{eq:response} \begin{array}{c} r19701291023892035483046984067960457640175310132100\\ r18970128102382035483046984067960457640175310132100\\ r1897012810238203548304698540679604576401745310132100\\ 193104239404954058650598506876091789024552013227100\\ 189210293203943075870498404854057650816780454012345100\\ r197012891023820354830459407960457640175101532100\\ 6196012891023782034730459870498408540576509167890123455100\\ r1970128910238203548304659407960457640175101552100\\ r197012910238203548304659406796045760174101452100\\ r19701291023820354830465806796045760174101452100\\ r19701291023892035483046984067960457601741014532100\\ r19701291023892035483046984067960457601741014532100\\ r19701291023892035830469854067960457601741014532100\\ r19701291023892035830469854059168703453012356100\\ r1970129102382035830469854067960457601741014532100\\ r1970129102382035830469854067960457601741014532100\\ r1970129102382035830469854067960457601741014532100\\ r19701281023820358304594554057507455401231013456100\\ r19201282032830358304874554505950274320134210245200\\ r197012810238203548304594059015675040175101532100\\ r1970128102382035483045940590575650157640175101532100\\ r192002320394307580704854067960457640175101532100\\ r19200232039430758070485405950575050157640175101532100\\ r192002320394307580704854055050501565001532102345200\\ r1970128102382035483048745550550501576023130132455100\\ r18970128102382035498304894067960457640175101532100\\ r189701280303383075807048940679504556015780027432015123013455100\\ r189701280303383049849406795005850615780027432013101345100\\ r1897012803233430498408940679500580157800274301310132500\\ r1897012803035830469840679500580157800254001755101532100\\ r189701280303383304898048940578058015780015780013501321020\\ r18970128030338304874057500580157800254500155015501052010321002\\ r189701283003300787078098740575058015780025450015501052050018850037550103020510027455010238500355000000000000000000000000000000$	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0	671602226016 671604206162 671604207016 671604207013 6716042207013 67160422003 67160422063 67160422063 671605006423 671605006423 671605006423 6716050044017 705141155501 72060371501 72060371501 721240364522 721240364522 721240365203
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058656509850687660917890245620132267100\\ 1892102932039430159870495405565091678091234561100\\ 7197012891023820354983046984067960457640175101532100\\ 619601289102326394301598704940679604576401175101532100\\ 7197012891023820354830469840679604576401175101532100\\ 7197012291023820354830469840679604576401175101532100\\ 71970129102382035483046984067960647601741014532100\\ 71970129102382035483046984067960647601741014532100\\ 71970129102382035483046984067960647601741014532100\\ 719701291023820354830469854067960647601741014532100\\ 719701291023820354830469854067960647601741014532100\\ 7197012910238203548304875406595059521733210622100\\ 712967071731037830384330485540576550595074743210672150\\71296023203943075870485405765091677902312113455100\\71297012291023820358830469854067960647601745310132100\\ 7197012891023820358830469854067960647601745310132100\\ 719701289102382035830469854067960647601745310132100\\ 18921029320394304754049874405755081676502354202455201256100\\7197012891023820358304698540679606476017455101321100\\ 18921029323039304030485440545667666047601745310132100\\ 189210293230393040304836404037406740576530345540373420123510321020\\ 1971701291203820358304698544660677860476017453131132100\\ 189210293234303043304640464060766474$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207013 671604227013 671604223007 671604223007 6716042242063 671604226017 671605004023 671605004017 705141155501 720261526700 721060371501 721240364522 721240364522 721240565203 721300670303
$\label{eq:second} \begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058650598506876609178902456201322671100\\ 18921029320394307598704954058650576509117890123456100\\ 1921029320394307587049540085405765091167800123456100\\ 1921029320394307587049540085405765091167800123456100\\ 17197012891023820255483046954067960457661175101532100\\ 6196012691023820255483046954067960457661175101532100\\ 7197012910238920354830469540677600647601175101532100\\ 192102932039430478540497405865091687903453012351010\\ 129102389203548304694656950476540175101235100\\ 17197012911023892035483046946569504765102131031452100\\ 7197012911023892035483046946569504765091342102452100\\ 71970121012910232039430483046946569504765091342102452100\\ 71297012810238203349830469465695069706047501175101532100\\ 71870128102382035493046946067960457640175101532110\\7187012810232203943047540498746576508167802345201256100\\ 71970128100238020354830469460576508167802345201256100\\ 719701291023920394304754049874655065608167602345201256100\\ 7197012910238920354303068606796606756601755001755101322100\\ 71870128100238020354830468660786067766067566017550017550132210234510125100\\7187012341023802035830668660786067766017550017550101322100\\718701291023807670969560677660175508167802345201256100\\7197012912030203430305870969560677660675608167660345500106721420122102342034201256100\\71$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207043 671604223007 671604242063 6716050046218 6716050046218 671605004423 6716050044017 705141155501 720261526700 721060371501 7212403645203 721240565103 721240665103 721301454303
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058656509850687660917890245620132267100\\ 1892102932039430159870495405565091678091234561100\\ 7197012891023820354983046984067960457640175101532100\\ 6196012891023262354983046984067960457640175101532100\\ 7197012891023820354430469540679604576401175101532100\\ 719701229102382035483046984067960457640175101532100\\ 71970129102382035483046984067960647601741014532100\\ 7197012910238220354830469854067960647601741014532100\\ 7197012910238220354830469854067960647601741014532100\\ 719701291023822035830469854067960647601741014532100\\ 719701291023822035830469854067960647601741014532100\\ 719701291023822035830469854067960647601741014532100\\ 719701231023822035830487540569505592574132106128102322039483075857640575590167703131345100\\ 71970128102232203548304698540679606476017455101322100\\ 7197012810223220354830469854067960647601745310132100\\ 71970128102320354830469854067960647601745310132100\\ 71970128102320354830469854067960647601745310132100\\ 71970128102320354830469854067960647601745310132100\\ 18921029323039430475404984640576560647601745310132100\\ 189210293230394304544065466067786047601745310132100\\ 189210293230394304754049846405765606476007145501351032102\\ 1971770128102330383030463046364045604756017453130123100\\ 18921029323039$	0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0	671602226016 671603016023 671604206162 671604207013 671604227013 671604223007 671604223007 6716042242063 671604226017 671605004023 671605004017 705141155501 720261526700 721060371501 721240364522 721240364522 721240565203 721300670303
$\label{eq:second} \begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058650598506876609178902456201322671100\\ 18921029320394307598704954058650576509117890123456100\\ 1921029320394307587049540085405765091167800123456100\\ 1921029320394307587049540085405765091167800123456100\\ 17197012891023820255483046954067960457661175101532100\\ 6196012691023820255483046954067960457661175101532100\\ 7197012910238920354830469540677600647601175101532100\\ 192102932039430478540497405865091687903453012351010\\ 129102389203548304694656950476540175101235100\\ 17197012911023892035483046946569504765102131031452100\\ 7197012911023892035483046946569504765091342102452100\\ 71970121012910232039430483046946569504765091342102452100\\ 71297012810238203349830469465695069706047501175101532100\\ 71870128102382035493046946067960457640175101532110\\7187012810232203943047540498746576508167802345201256100\\ 71970128100238020354830469460576508167802345201256100\\ 719701291023920394304754049874655065608167602345201256100\\ 7197012910238920354303068606796606756601755001755101322100\\ 71870128100238020354830468660786067766067566017550017550132210234510125100\\7187012341023802035830668660786067766017550017550101322100\\718701291023807670969560677660175508167802345201256100\\7197012912030203430305870969560677660675608167660345500106721420122102342034201256100\\71$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207043 671604223007 671604242063 6716050046218 6716050046218 671605004423 6716050044017 705141155501 720261526700 721060371501 7212403645203 721240565103 721240665103 721301454303
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$		671602226016 671603016023 671604207016 671604207018 6716042207043 67160422007 67160422003 67160422003 671604242063 671605006216 67160500423 671605004423 671605004423 77160504423 671605044017 705141155501 720261526700 721060371501 721240565203 72124056503 721240665103 721301454303 721301454303 72141525603
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207018 671604242063 671604242063 671605004023 671605004023 671605004023 671605004017 705141155501 720261526700 721060371501 72124056203 721240565103 721240565103 721301454303 7214124225603
$\begin{tabular}{lllllllllllllllllllllllllllllllllll$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207018 671604242063 671604242063 671605004023 671605004023 671605004023 671605004017 705141155501 720261526700 721060371501 72124056203 721240565103 721240565103 721301454303 721412425603
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058656509850687660917490245620132267110\\ 1892102932039430159870495405565091678901234561100\\ 19211029203944307587049840657650916178901234561100\\ 719701289102382035443046944067960457640175101532100\\ 61960128910231820347304998440569504654011451114321100\\ 719701229102382034430469406796044760174111014532100\\ 719701291023802035483046984067960447601751011532100\\ 71970129102380203548304698406796044760174110145321100\\ 6189601281023822035483046985406796045765014741014532100\\ 712967701731031830384330489854065950595217432016422100\\ 7129671731031830384330489854065765901647601741014532100\\ 7129677012310238203549830469854067960457640175101532100\\ 71970128102382035498304698540679604576401755101532100\\ 71970128102320394330475404984067960457601745310132100\\ 189210293203943047540498476540575508167802342012455100\\ 71970128102320394330475404984765606756091678002342012456100\\ 8128029820234492046940678601755015321103456100\\ 8128029820234420469406786017550016780023420124556100\\ 8128029820234420469406786017650016780023420124556100\\ 1921029320394830475404874005765091678901231013456100\\ 1921029320394830475404874057550916789034331013456100\\ 1921029320394830475404874057550916789034331013456100\\ 1921029320394830475404874057550916789$		671602226016 671603016023 671604206162 671604207018 671604207018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240565203 721240665103 721301454303 7214124222714
$\label{eq:second} \begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239409540586505985068766917890245620132671100\\ 1892102932039430758704954058506576509117800123456100\\ 19210293203943075870495400455405509167890123455100\\ 6196012891023728203548304698406796045766101751015321100\\ 619601289102372820354730469540679604576611410114521100\\ 71970128910238203548304695406796045766117410145321100\\ 71970128910238203548304695406796045766117410145321100\\ 7197012910239420354830469540679604576617410114521100\\ 7197012910239420354830469540679604576617410114521100\\ 7197012910239220354830469540679604576401741014521100\\ 712970173103783038430487654056950595025720134210234210245200\\ 7197012810238203584304695406796045601745301123100\\ 71970128102382035843046954067960456017550017453012310132100\\ 1892102932039430758704497405765081678002342012256100\\ 7197012910239420354830469406789601756500175310132100\\ 18921029320394304754049840678604576500155310132100\\ 1892102932039430475404984067860457650017550015321035430147100\\ 18210293203943047540487405765091678902342012351013456100\\ 71970129102392039430475404874005765091678901231013456100\\ 19210293203943047540487405765091678900353013356100\\ 19210293203943047540487405765091678900353013356100\\ 19210293203943047540487405765091678900353013356100\\ 192102932039430475404874057650916789003530133551100\\ 19210293203$		671602226016 671604206162 671604207013 671604207013 6716042207013 6716042207043 67160422007 67160422007 671605006216 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 72124056503 721301454303 721301454303 72144222714 72144224703
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 19210292203943075987049540576509167890123456110\\ 19210292203943075987049540576509167890123456110\\ 7197012891023820354983046984067960457640175101332100\\ 6196012891023720394301598744056950465400175101532100\\ 719701289102382035483046984067960457640175101532100\\ 719701229102382035483046954067960457640175101532100\\ 71970129102382035483046954067960647601741014532100\\ 719701291023802035483046984067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 71970129102380203548304698540679606476001741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 719701231013423020354830487654057650971678901231013455100\\ 71970128102232203943075404983467565057550017550115321032100\\ 189210293203943075404987405765091678901231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404986405765091678001231013455100\\ 1921029320394307540498645550575590167850312350123551105521105521105521105521105521105521105521105521105521105521105521$		671602226016 671603016023 671604206162 671604207018 671604207018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240565203 721240665103 721301454303 7214124222714
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 19210292203943075987049540576509167890123456110\\ 19210292203943075987049540576509167890123456110\\ 7197012891023820354983046984067960457640175101332100\\ 6196012891023720394301598744056950465400175101532100\\ 719701289102382035483046984067960457640175101532100\\ 719701229102382035483046954067960457640175101532100\\ 71970129102382035483046954067960647601741014532100\\ 719701291023802035483046984067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 71970129102380203548304698540679606476001741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 719701231013423020354830487654057650971678901231013455100\\ 71970128102232203943075404983467565057550017550115321032100\\ 189210293203943075404987405765091678901231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404987405765091678001231013455100\\ 19210293203943075404986405765091678001231013455100\\ 1921029320394307540498645550575590167850312350123551105521105521105521105521105521105521105521105521105521105521105521$		671602226016 671604206162 671604207013 671604207013 671604227043 6716042207043 67160422007 671604242063 671605004023 671605004023 671605004023 671605044017 705141155501 720261526700 72060371501 721240364522 721240565203 721240565103 721301454303 721301454303 721442222714 721442222714 72144222144
$\label{eq:second} \begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 19310423940495405865059850687609178902455201322671100\\ 1892102932039430159870495405765091678001234561100\\ 19211023203943015987049400654565091578090123451100\\ 19210232039430159870494006549604576401175101532100\\ 619601289102382035483046984065960455640175101532100\\ 61960128910238203548304698406796045764011751015321100\\ 192102320394303473049540659504654011751015321100\\ 19210232039430347830469410679604576401751015321100\\ 6189601280102380203548304698540679606476017410145321100\\ 61896012801023802035483046985406579504765402743201672100\\ 71297017310031830384330489454545659505765001745101321010\\ 8189601280102820234983045940657650916789012310134554100\\ 7189701281102380203548304698406796064756001753101321000\\ 189210293203943075870489440679606476001753101321000\\ 1892102932039430075804985405765091678901231013456100\\ 719701281023202344304754048740575508167802342012456100\\ 812802982023443047580487405755081678902342012456100\\ 81280298202344304758048740575508167890234212456100\\ 812802982023443047580487405755081678901231013456100\\ 719701289102382035833047560617650916789035301231013456100\\ 719701289102382035833047560617650916789035301231013456100\\ 719701289102382035833047560617650916769035301235101356100\\ 7197012891023820358330475606765091167690175510153510135010\\ 719701281$		671602226016 671603016023 671604207016 671604207018 6716042207018 67160422007 67160422003 67160422003 671605006216 67160500423 671605004017 705141155501 720261526700 721060371501 721240565203 72140364522 72124056503 72130165203 7214124222714 721442222714 7214422231303 7214422231303 721443024703 72556020423 740522515114
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 192102920394301598704954055609117890234561100\\ 192102920394301598704954057650916778001234561100\\ 719701289102382035498304698406796044760175101532100\\ 619601289102382035498304698406796044760175101532100\\ 71970128910238203548304695406796064760017410143521100\\ 71970128910238203548304695406796064760017410143521100\\ 719701299102389203548304695406796064760017410145521100\\ 71970212910238920354830469540679606476017410145521100\\ 618960128102380203548304696540679606476017410145521100\\ 618960128102380203548304696540679606476017410145521100\\ 719702129102382035483046945456595025920134210245200\\ 192102932039430475540485405765091678901231013455100\\ 7197012810238020354830469465765091678901231013455100\\ 192102932039430175500487405765091678901231013455100\\ 192102932039430175500487405765091678901231013455100\\ 19210293203943017540487405765091678903453012356100\\ 19210293203943017540487405765091678903453012356100\\ 19210293203943017540487405765091678903453012356100\\ 19210293203943017540487405765091678903453012356100\\ 192102932039430047540487405765091678903453012356100\\ 192102932039430047540487405765091678903453012356100\\ 192102932039430047540487405765091678903453012356100\\ 192102932039430047540487405765091678903453012356100\\ 192102932039430$		671602226016 671604206162 671604207013 671604207013 671604227013 67160422007 671604242063 671604242063 671605004023 671605004023 671605004017 705141155501 720261526700 721060371501 7214152501 721440565203 721240665103 721240665103 721301454303 721442221714 721442231303 721442221714 721442221714 721442221714 721442221714 721442221714 721442221714 7214422215114
$\label{eq:second} \begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 19310423940495405865059850687609178902455201322671100\\ 1892102932039430159870495405765091678001234561100\\ 19211023203943015987049400654565091578090123451100\\ 19210232039430159870494006549604576401175101532100\\ 619601289102382035483046984065960455640175101532100\\ 61960128910238203548304698406796045764011751015321100\\ 192102320394303473049540659504654011751015321100\\ 19210232039430347830469410679604576401751015321100\\ 6189601280102380203548304698540679606476017410145321100\\ 61896012801023802035483046985406579504765402743201672100\\ 71297017310031830384330489454545659505765001745101321010\\ 8189601280102820234983045940657650916789012310134554100\\ 7189701281102380203548304698406796064756001753101321000\\ 189210293203943075870489440679606476001753101321000\\ 1892102932039430075804985405765091678901231013456100\\ 719701281023202344304754048740575508167802342012456100\\ 812802982023443047580487405755081678902342012456100\\ 81280298202344304758048740575508167890234212456100\\ 812802982023443047580487405755081678901231013456100\\ 719701289102382035833047560617650916789035301231013456100\\ 719701289102382035833047560617650916789035301231013456100\\ 719701289102382035833047560617650916769035301235101356100\\ 7197012891023820358330475606765091167690175510153510135010\\ 719701281$		671602226016 671603016023 671604207016 671604207018 6716042207018 67160422007 67160422003 67160422003 671605006216 67160500423 671605004017 705141155501 720261526700 721060371501 721240565203 72140364522 72124056503 72130165203 7214124222714 721442222714 7214422231303 7214422231303 721443024703 72556020423 740522515114
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058650598506876091789024562013227110\\ 18921029320394307587049540576598104780024562013227100\\ 19210293203943075870498408580576509167890123456100\\ 71970128910238203447304598076509164780123456100\\ 7197012891023820344730459804654067960457640175101532100\\ 6196012891023703943046954067960457640175101532100\\ 6196012891023703943046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046984067960457611741014532100\\ 7197071291023802035483046985406796045761741014532100\\ 7197071291023802035483046985406796045761741014522100\\ 712967071731037840348340556556591678901231013456100\\ 7189701281023820354983045940576509167802345201252100\\ 7197012891023802035483046985406796045761745310132100\\ 18920023203943075404987405765091678091231013456100\\ 71897012891023820358904698540679604576041755101532100\\ 719701289102382035890469854057650916780032345202345200\\ 71970128910238203583046965405756091678003453012356100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678091231013456100\\ 1921029320398430475404974555091678091231013456100\\ 1921029320398430475404974555091678091231013456100\\ 71970128910238203583046985405765091678091231013456100\\ 71970128910238203583046965505765091678001231013456100\\ 192102932039843047540487405765091678091231013456100\\ 71970128910238203583046965505765091678091231013456100\\ 719701289102382039843047540487405765091678091231013456100\\ 7197012891023820398430475404874057650916780123452100\\ 7187012891023820398430475404874057650916780123452100\\ 7187012801023820398430475404874057550816780123456100\\ 718701280102382039843047540487405750816780123456100\\$		671602226016 671604206162 671604207018 671604207018 671604227018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240665103 721240665103 721442227144 72144222714 72144222714 72144222714 72144222714 72144222714 721442221503 72144222515114 741501424703 74150220523
$\begin{array}{c} 719701291023802035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 193104239404954058650598506876091789024562013267100\\ 18921029320394307587049540576508167804540175101532100\\ 6192102932039430758704984048540576508167804540175101532100\\ 6196012891023762034730459807647960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701280223692035483046954067960457640175101532100\\ 719202320394307587048540569504765402743201672100\\ 712967017310378038430487654056950259201342102345200\\ 719701280102382035883046954067960457640175310132100\\ 18921029320394307547048540576508167802345201256100\\ 71970128010238203588304698540576508167802345201256100\\ 719701280102382035883046985405765081678033530123501032100\\ 1892102932039430754704985405765081678003453012350125010\\ 719701280102382035883046960576509167800345301231013456100\\ 71970128010238203588304696057650916780034530123501032100\\ 1892102932039430475404874057650916780034530123501235010\\ 71970128010238203588304696057650916780034530123501032100\\ 18921029320398430754004796057650916780034530123501032100\\ 18921029320398430754704954057650916780034530123501032100\\ 192102932039843075400479605765091678003453012350100\\ 7197012810238203588304698506796047640174531013456100\\ 7197012810238203588304756509565950576509167800353012350100\\ 19210293203984307540047565091678003453012350100\\ 19210293203984304754048705765091678003453012350100\\ 7197012810238203583056950545506796047640174532100\\ 967907187012810238203598305695054550576509167800235010234100\\ 7189071870128102382$		671602226016 671604206162 671604207018 671604207018 6716042207018 6716042207018 67160422007 67160422007 671605006218 67160500423 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240565103 721301454303 72144222714 721442231503 72144222714 72144224703 72144204703 741501424703 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 74150224523
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 193104239404954058650598506876091789024562013227110\\ 18921029320394307587049540576598104780024562013227100\\ 19210293203943075870498408580576509167890123456100\\ 71970128910238203447304598076509164780123456100\\ 7197012891023820344730459804654067960457640175101532100\\ 6196012891023703943046954067960457640175101532100\\ 6196012891023703943046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046984067960457611741014532100\\ 7197071291023802035483046985406796045761741014532100\\ 7197071291023802035483046985406796045761741014522100\\ 712967071731037840348340556556591678901231013456100\\ 7189701281023820354983045940576509167802345201252100\\ 7197012891023802035483046985406796045761745310132100\\ 18920023203943075404987405765091678091231013456100\\ 71897012891023820358904698540679604576041755101532100\\ 719701289102382035890469854057650916780032345202345200\\ 71970128910238203583046965405756091678003453012356100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678003234521256100\\ 71970128910238203583046965405756091678091231013456100\\ 1921029320398430475404974555091678091231013456100\\ 1921029320398430475404974555091678091231013456100\\ 71970128910238203583046985405765091678091231013456100\\ 71970128910238203583046965505765091678001231013456100\\ 192102932039843047540487405765091678091231013456100\\ 71970128910238203583046965505765091678091231013456100\\ 719701289102382039843047540487405765091678091231013456100\\ 7197012891023820398430475404874057650916780123452100\\ 7187012891023820398430475404874057650916780123452100\\ 7187012801023820398430475404874057550816780123456100\\ 718701280102382039843047540487405750816780123456100\\$		671602226016 671604206162 671604207018 671604207018 671604227018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240665103 721240665103 721442227144 72144222714 72144222714 72144222714 72144222714 72144222714 721442221503 72144222515114 741501424703 74150220523
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 1921029220394307598704954057650916780023456110\\ 192102922039430759870495405765091678090123456110\\ 7197012891023820354983046984067960457640175101532100\\ 61960128910237203943045981045964067960457640175101532100\\ 719701289102382035483046984067960457640175101532100\\ 71970129102382035483046954067960467600171410145321100\\ 719701291023820354830469540679606476017410145321100\\ 719701291023820354830469840679606476017410145321100\\ 71970129102382035483046984067960647600174101145321100\\ 6189601281023822055483046985406796064760174101145321100\\ 6189601281023822035483046985406796064760174101145321100\\ 712967071731037830384330487554059505575501153210152210621210232203943047540498104967496045765091678901231013456100\\ 719701281023820358830469854067960647601745310132100\\ 1892102932039430475404987405765091678091231013456100\\ 192102932039430475404984067565091678091231013456100\\ 19210293203943047540487405765091678091231013456100\\ 71970128102382035830469654067765091678091231013456100\\ 19210293203943047540487405765091678091231013456100\\ 19210293203943047540487405765091678091231013456100\\ 71870712801230233230363036553036555055565657667960476076575212551053212356100\\ 718707121027320394304754048740576509116780723522100\\ 1891012210293203983034754048$		671602226016 671604206162 671604207018 671604207018 671604227018 6716042207018 671604242063 671604242063 671605004017 705141155501 720261526700 72060371501 721240364522 721240565203 721240565103 721301454303 721301454303 721412222714 721442222714 721442222714 7214422231303 721442222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 721442024703 72556020423 741502424523 741502424523 741502424523 741502424523
$\begin{array}{c} 719701291023802035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 193104239404954058650598506876091789024562013267100\\ 18921029320394307587049540576508167804540175101532100\\ 6192102932039430758704984048540576508167804540175101532100\\ 6196012891023762034730459807647960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701280223692035483046954067960457640175101532100\\ 719202320394307587048540569504765402743201672100\\ 712967017310378038430487654056950259201342102345200\\ 719701280102382035883046954067960457640175310132100\\ 18921029320394307547048540576508167802345201256100\\ 71970128010238203588304698540576508167802345201256100\\ 719701280102382035883046985405765081678033530123501032100\\ 1892102932039430754704985405765081678003453012350125010\\ 719701280102382035883046960576509167800345301231013456100\\ 71970128010238203588304696057650916780034530123501032100\\ 1892102932039430475404874057650916780034530123501235010\\ 71970128010238203588304696057650916780034530123501032100\\ 18921029320398430754004796057650916780034530123501032100\\ 18921029320398430754704954057650916780034530123501032100\\ 192102932039843075400479605765091678003453012350100\\ 7197012810238203588304698506796047640174531013456100\\ 7197012810238203588304756509565950576509167800353012350100\\ 19210293203984307540047565091678003453012350100\\ 19210293203984304754048705765091678003453012350100\\ 7197012810238203583056950545506796047640174532100\\ 967907187012810238203598305695054550576509167800235010234100\\ 7189071870128102382$		671602226016 671604206162 671604207018 671604207018 6716042207018 6716042207018 67160422007 67160422007 671605006218 67160500423 67160500423 671605004017 705141155501 720261526700 721060371501 721240364522 721240565203 721240565203 721240565103 721301454303 72144222714 721442231503 72144222714 721442231503 72144224703 741501424703 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 741502230523 74150224523
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 192102920394301598704954055609117890245620132267100\\ 1892102932039430159870495405565691678001234561100\\ 719701289102382035498304698406796044760175101532100\\ 619601289102382035498304698406796044760175101532100\\ 71970128910238203548304699840679604476017410145221100\\ 71970128910238203548304699840679604476017410145221100\\ 719701291023802035483046998406796064760017410145321100\\ 719701291023802035483046998406796064760017410145321100\\ 61896012810238020354830469840679606476017410145321100\\ 618960128102380203548304698540679606476017410145321100\\ 6189601281023802035483049804054950550559016772031121252100\\ 7197017131037130303830449854965655055950525210672197212232203948304754046746057650916789012310133456100\\ 71970128102380203548304494605765091678901231013456100\\ 7197012810238020354830447405765091678901231013456100\\ 19210292023443047540487405765091678901231013456100\\ 19210293203943047540487405765091678903453012356100\\ 71970128912023203548304754048740576509167890012310134561100\\ 192102932039430475404874057650916789001231013456100\\ 192102932039430475404874057650916789001231013456100\\ 71970128912232233948304754048740576550916780012351105321100\\ 18911021102210229223349330475404874057655091678001235251100\\ 719701281023203643304754048740$		671602226016 671604206162 671604207018 671604207018 671604227018 6716042207018 671604242063 671604242063 671605004017 705141155501 720261526700 72060371501 721240364522 721240565203 721240565103 721301454303 721301454303 721412222714 721442222714 721442222714 7214422231303 721442222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 721442024703 72556020423 741502424523 741502424523 741502424523 741502424523
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 192102920394301598704954055609117890245620132267100\\ 1892102932039430159870495405565691678001234561100\\ 719701289102382035498304698406796044760175101532100\\ 619601289102382035498304698406796044760175101532100\\ 71970128910238203548304699840679604476017410145221100\\ 71970128910238203548304699840679604476017410145221100\\ 719701291023802035483046998406796064760017410145321100\\ 719701291023802035483046998406796064760017410145321100\\ 61896012810238020354830469840679606476017410145321100\\ 618960128102380203548304698540679606476017410145321100\\ 6189601281023802035483049804054950550559016772031121252100\\ 7197017131037130303830449854965655055950525210672197212232203948304754046746057650916789012310133456100\\ 71970128102380203548304494605765091678901231013456100\\ 7197012810238020354830447405765091678901231013456100\\ 19210292023443047540487405765091678901231013456100\\ 19210293203943047540487405765091678903453012356100\\ 71970128912023203548304754048740576509167890012310134561100\\ 192102932039430475404874057650916789001231013456100\\ 192102932039430475404874057650916789001231013456100\\ 71970128912232233948304754048740576550916780012351105321100\\ 18911021102210229223349330475404874057655091678001235251100\\ 719701281023203643304754048740$		671602226016 671604206162 671604207018 671604207018 671604207018 6716042207018 67160422007 67160422007 671605006218 671605004023 671605004017 705141155501 720261526700 721060371501 721441552701 721240565203 721240565203 721240565103 721240565103 721301454053 7213412425603 721442222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222714 72144222515114 741502230523 74150224523 74150224523
719701291023892035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         1937014239404954058650598506876091789024562013267100         189210293203943075987049540576509167890124562013267100         189210293203943075987049540576509167890124565100         719701289102382035498304694067960457640175101532100         61960128910237820347304998408585076509167890123456100         7197012910238203548304694067960457640175101532100         6196012891023782034730499840657960457640175101532100         719701291023892035483046984067960457640175101532100         719701291023892035483046984067960457640175101532100         7197012910238920354830469854067960457640175101532100         71970129102389203548304698540679604576401741014532100         71970129102389203588304698540659505292014820134210245200         7197012910238920354830469854065765091678901231013456100         718970128102382035498304698540679604576401755101532100         71970128102382035498304698540679604576401755101532100         7197012801023820358304698540679604576401755101532100         71970128010238203583046965565905167801745310132100         189210293203984304754048740576508167802345201256100         718701280102382035830469645765091678901231013456100	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0	671602226016 671604206162 671604207018 671604207018 671604227018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 72060371501 721240364522 721240565203 72124056503 721301454303 721301454303 721442222714 721442222714 7214422231303 721442222714 72144222315114 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424163 741503050217 745542120145
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 71897012810238203548304698540679604576401745310132100\\ 193104239404954058650598506876091789024562013267100\\ 189210293203943075870498404854057650816780454012346100\\ 192102932039430758704984048540576509167890123452100\\ 619601289102378203473045980740984067960457611741014532100\\ 719701289102382023548304695406796045761174101452100\\ 7197012891023820235483046954067960457611741014522100\\ 719701291023892035483046954067960457640175101532100\\ 619601289102382035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701280102382034983045940569504764007741014552100\\ 71296701731037803843048765405695047640175101532100\\ 718701280102382035883046954067960457640175101532100\\ 7187012801023820358830469540679604576401755310132100\\ 189210293203943075470485405765081678001231013456100\\ 71970128010238203588304694067860457640175310132100\\ 1892102932039430754704965405765081678001231013456100\\ 719701280102382035883046965165765091678901231013456100\\ 719701280102382035883045960176510153210356430187100\\ 1921029320394304754048740576509167890345301132100\\ 1892102932039430475404874057650916789034530123501032100\\ 1892102932039430475404874057650916789034530123501032100\\ 719701280102382035883046965405765091678903453012350100\\ 719701280102382035883046940679605765091678903453012350100\\ 71970128010238203588304594056756091678903453012356100\\ 71970128010238203588304594056756091678004576001741014552100\\ 94590228202349230484304754048740576509167890123450100\\ 71897018801238203588304594056950545067960476401745321200\\ 94590229203984304754048740576509167800123410014234100\\ 94590229203984304754048740576509167800123450100\\ 718970128102238203584304754048740576509167800123450100\\ 718970128102238203584304754048$		671602226016 671604206162 671604207013 671604207013 6716042207013 6716042207013 67160422007 67160422007 67160500420017 6716050042017 705141155501 720261526700 721040371501 721240364522 721240364522 721240565103 72130145303 72130145303 721442221714 721240565103 72130145303 721442231303 721442231303 7214422501 7214422515114 741501424703 74150220523 74150244163 7415022017
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 192102922039430759870495405765091677800234561100\\ 192102922039430759870495405765091677800123456100\\ 7197012891023820354983046984067960457640175101532100\\ 619601289102372039430459810459640679604576401175101532100\\ 7197012891023820354830469840679604576401175101532100\\ 71970129102382035483046954067960465401175101532100\\ 71970129102382035483046984067960467601741014532100\\ 7197012910238020354830469840679606476017410114532100\\ 7197012910238020354830469854067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 712967701731003183038433048755405950557052743201672100\\ 712907173100318034830469854065765091678901231013455100\\ 71970128910232039430475404987405765091678001231013455100\\ 71970128910232039430475404984405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 719701289102320394304754048740576509167800123103455100\\ 719701289102320394304754048740576509167800123103455100\\ 719701289102320394304754048740576509167800123103455100\\ 7197012810229230394304754048740576509167800123103455100\\ 718707187012810238220358330475$	0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 1 0 0 0 0	671602226016 671604206162 671604207018 671604207018 671604227018 67160422007 671604242063 671604242063 671605004017 705141155501 720261526700 72060371501 721240364522 721240565203 72124056503 721301454303 721301454303 721442222714 721442222714 7214422231303 721442222714 72144222315114 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424523 741502424163 741503050217 745542120145
$\begin{array}{c} 719701291023820235483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 192102922039430759870495405765091677800234561100\\ 192102922039430759870495405765091677800123456100\\ 7197012891023820354983046984067960457640175101532100\\ 619601289102372039430459810459640679604576401175101532100\\ 7197012891023820354830469840679604576401175101532100\\ 71970129102382035483046954067960465401175101532100\\ 71970129102382035483046984067960467601741014532100\\ 7197012910238020354830469840679606476017410114532100\\ 7197012910238020354830469854067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 7197012910238020354830469854067960647601741014532100\\ 712967701731003183038433048755405950557052743201672100\\ 712907173100318034830469854065765091678901231013455100\\ 71970128910232039430475404987405765091678001231013455100\\ 71970128910232039430475404984405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 7197012891023203943047540487405765091678001231013455100\\ 719701289102320394304754048740576509167800123103455100\\ 719701289102320394304754048740576509167800123103455100\\ 719701289102320394304754048740576509167800123103455100\\ 7197012810229230394304754048740576509167800123103455100\\ 718707187012810238220358330475$		671602226016 671604206162 671604207015 671604207015 671604227016 6716042207015 671604242063 671604242063 671605004023 671605004023 671605004023 671605004023 705141155501 720261526700 721060371501 72140364522 721240565203 721240565103 721442222714 721442231303 721442222714 721442231303 721442222714 721442231303 721442222714 721442231303 721442222714 721442231303 721424065103 721424065103 721442222714 721442231303 7214222515114 741502230523 74150224523 74150224523 741502244163 74150360217 745543010213 750537000423
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 193104239404954058650598506876091789024552013267100\\ 18921029320394307587049540576598167804546012346100\\ 192102932039430758704984068540576509167890123456100\\ 719701289102382035483046984067960457640175101532100\\ 6196012891023782034730495405960647601741014532100\\ 719701291023802035483046954067960457640175101532100\\ 61960128910237820347304954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 7197012910238020354830469854067960457640175101532100\\ 71970129102380203548304698540679604576401741014532100\\ 7197012910238020354830469854057960457601741014532100\\ 719701291023802035483046985405796045760174101532100\\ 8189601281023820349830459405695094765402743201672100\\ 71296707173103780384540497605565091678901231013456100\\ 7197012891023802035483046985405765091678901231013456100\\ 7197012801023802035483046985405765091678901231013456100\\ 71970128020394307580748540576508167802342012456100\\ 81280298202349200354830449405765091678901231013456100\\ 7197012890203548304754049740576508167802342012456100\\ 8128029820234920035483047560176510153210556430147100\\ 189210293203984304754048740576508167802342012456100\\ 8128029820234920454930475870496545067960476017531013456100\\ 719701289102382035830475904760576508167802342012456100\\ 812802982023492045943047565091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 719701289102382035830459405765091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 7187071870128102382035830459405765091678901231013456100\\ 718707187012810238203583045940576509167890135101321055430147100\\ 9657901718701281023820358830459405765091678901235012356100\\ 71870788071870128102382035483045540487405765091678903153012356100\\ 7187078807128102$		671602226016 671604206023 671604207043 671604207043 6716042207043 67160422007 67160422003 671604242063 671605006216 671605004017 705141155501 720261526700 721060371501 721240565203 7214025701 72124056503 7214056503 72141242503 721442222714 7214422231303 721442222714 7214422231303 721442223133 721442223133 72144222313133 72144222313133 72144222313133 72144222313133 72144222313133 72144222313133 721452502242 72145203233 741502244163 74150244163 745542120145 745542120145 745542120145 745542120145 745543010213 750725100216
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 19210292039430158650598506876091789024562013227110\\ 1892102932039430159870495405565091678001234561100\\ 719701289102382035498304698406796045764017550123456110\\ 7197012891023820354983046984067960457640175101532100\\ 6196012891023720234430459874405695014654011741014521100\\ 719701289102382035483046984067960447601741014521100\\ 719701291023802035483046984067960447601741014532100\\ 719701291023802035483046984067960647601741014532100\\ 719701291023802035483046984067960647601741014532100\\ 61896012810232023498304594065495059505920174201672100\\ 6189601281023202354833046984067960647601745310132100\\ 12206223239430475810485405675091678901231013455100\\ 71970128102232203943047540497405755081678002342012455100\\ 7197012810238023543014985405755091678901231013455100\\ 19210293203943047540487405765091678901231013456100\\ 19210293203943047540487405765091678901231013456100\\ 19210293203943047540487405765091678903450115310132100\\ 1892102932039430475404874057650916789034501153101321100\\ 1821022932039430475404874057650916789034501153101321100\\ 192102932039430475404874057650916789034501153101321100\\ 7187071297012810232035943047540487405765091678903453012356100\\ 718707129701281023203594304754048740576509167890345301235450100\\ 718707128020328430475404874$		671602226016 671604206162 671604207018 671604207018 6716042207018 6716042207018 67160422007 67160422007 67160500423 67160500423 705141155501 720261526700 721060371501 721240565203 721240565203 721240565103 721240565103 721301454053 721240565103 721301454053 721443024703 72144222714 72144222714 72144222714 72144224703 7214256020423 74150224523 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 750725100210 75072510020
$\begin{array}{c} 719701291023892035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 7189701281023820354830469854067960457640175310132100\\ 193104239404954058650598506876091789024552013267100\\ 18921029320394307587049540576598167804546012346100\\ 192102932039430758704984068540576509167890123456100\\ 719701289102382035483046984067960457640175101532100\\ 6196012891023782034730495405960647601741014532100\\ 719701291023802035483046954067960457640175101532100\\ 61960128910237820347304954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 719701291023802035483046954067960457640175101532100\\ 7197012910238020354830469854067960457640175101532100\\ 71970129102380203548304698540679604576401741014532100\\ 7197012910238020354830469854057960457601741014532100\\ 719701291023802035483046985405796045760174101532100\\ 8189601281023820349830459405695094765402743201672100\\ 71296707173103780384540497605565091678901231013456100\\ 7197012891023802035483046985405765091678901231013456100\\ 7197012801023802035483046985405765091678901231013456100\\ 71970128020394307580748540576508167802342012456100\\ 81280298202349200354830449405765091678901231013456100\\ 7197012890203548304754049740576508167802342012456100\\ 8128029820234920035483047560176510153210556430147100\\ 189210293203984304754048740576508167802342012456100\\ 8128029820234920454930475870496545067960476017531013456100\\ 719701289102382035830475904760576508167802342012456100\\ 812802982023492045943047565091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 719701289102382035830459405765091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 7197012891023820358304759045765091678901231013456100\\ 7187071870128102382035830459405765091678901231013456100\\ 718707187012810238203583045940576509167890135101321055430147100\\ 9657901718701281023820358830459405765091678901235012356100\\ 71870788071870128102382035483045540487405765091678903153012356100\\ 7187078807128102$		671602226016 671604206023 671604207043 671604207043 6716042207043 67160422007 67160422003 671604242063 671605006216 671605004017 705141155501 720261526700 721060371501 721240565203 7214025023 72124056503 72124056503 721240665103 721301654303 721412422014 72141242503 721442222714 72144222714 7214222714 7214222714 7214222714 72144222714 72144222714 72144222714 72144222714 72144222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 7214222714 721422714 7214222714 7214222714 7214222714 721422714 7214222714 7214222714 7214222714 7214222714 721422714 721422714 721471727 7214577727 721271777777777777777
719701291023892035483046984067960457640175310132100         7189701281023820354830469854067960457640175310132100         7189701281023820354830469854067960457640175310132100         1937012810238203548304698540679604576401745310132100         193104239404954058650598506876091789024562013267100         189210293203943075987049540576509167890124562013227100         189210293203943075987049540576509167860912456100         719701289102382035498304698046796045764017540123456100         71970128910238203549830469840679604576401754011532100         619601228910238203548304698406796045764017540175101532100         71970129102389203548304698406796045764017540175101532100         719701291023802035483046985406796064760174101453012356100         719701291023892035483046985406796064760174101453012356100         7197012910238920354830469854067960647601741014532100         61896012810238020354830469854067960647601741014532100         71970128102380203548304498540559505476550213201672100         6189601281023820358030469854067960457640175510132100         719701281023802035483044984057860457640175510132100         719701281023802035483044984057860457640175510132100         71970128010238020354830449840678604576401755101322100         719701280203943047540487405765091678023452012545010         719701280203943047540487405765091678003453012356430187100             7197012802039483047540487405765091678003453012356430187100		671602226016 671604206162 671604207013 671604207013 671604227003 671604242063 671604242063 671604242063 671605004017 705141155501 720261526700 72060371501 72140505203 721240565203 721240565103 721301454303 721301454303 72141222714 721442222714 721442222714 7214422231303 721442222714 721442222714 7214422231303 72144224703 725542020423 741502424523 741502424523 741502424523 741502424523 741502424523 741503050217 745542120145 75537000423 750725100216 751301262007 751522120145
719701291023892035483046984067960457640175310132100         7189701281023820354830469854067960457640175310132100         7189701281023820354830469854067960457640175310132100         1937012810238203548304698540679604576401745310132100         193104239404954058650598506876091789024562013267100         189210293203943075987049540576509167890124562013227100         189210293203943075987049540576509167860912456100         719701289102382035498304698046796045764017540123456100         71970128910238203549830469840679604576401754011532100         619601228910238203548304698406796045764017540175101532100         71970129102389203548304698406796045764017540175101532100         719701291023802035483046985406796064760174101453012356100         719701291023892035483046985406796064760174101453012356100         7197012910238920354830469854067960647601741014532100         61896012810238020354830469854067960647601741014532100         71970128102380203548304498540559505476550213201672100         6189601281023820358030469854067960457640175510132100         719701281023802035483044984057860457640175510132100         719701281023802035483044984057860457640175510132100         71970128010238020354830449840678604576401755101322100         719701280203943047540487405765091678023452012545010         719701280203943047540487405765091678003453012356430187100             7197012802039483047540487405765091678003453012356430187100		671602226016 671604206162 671604207013 671604207013 6716042207013 6716042207013 67160422007 67160422007 6716050042017 705141155501 720261526700 721060371501 721240364522 721240364522 721240565203 72130145303 72130145303 72130145303 721443024703 725250020423 7415022015114 74150244163 74150220145 745543010213 750725100216 751522120145 75152210423
719701291023802035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         193704239404954058650598506876091789024562013227100         189210293203943075987049540576598106786091789024562013227100         189210293203943075987049540576509167890123456100         7197012891023820354983046940679604576509167890123456100         719701289102382035498304694067960457640175101532100         6196012891023782034730499840657960476017410145321200         71970129102382035983046954067960457640175101532100         719701291023802035483046984067960457640175101532100         719701291023802035483046984067960457601741014532100         719701291023802035483046984067960457640175101532100         71970129102380203548304698540679604576401741014532100         7197012910238020354830469854065765091678901231013456100         71970128102382035498304594056765091678001231013456100         7197012810238203549830469655675091678001231013456100         7187012810238203549830469645765091678001231013456100         719701280203983047540497405765091678001231013456100         719701280102382035830469645765091678001231013456100         719701280102382035830469645765091678001231013456100         719701280102382035830469607765091678001231013456100         719701280102382035830469605765091678001231013456100               719701280102382		671602226016 671604206162 671604207015 671604207015 671604227016 6716042207015 671604242063 671604242063 671605004023 671605004023 671605004023 720261526700 720261526700 721060371501 72141155501 72140565203 721240565203 721240565103 72140565203 721442221714 721442231303 72144222714 721442231303 721442221714 721442231303 721442024703 7215256020423 741502244523 741502244533 74150224523 74150224523 74150224523 74150224103 745542120145 755537000423 750725100216 751522120423 751522102423
$\begin{array}{c} 719701291023802035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 71897012810238203548304698540679604576401745310132100\\ 1921029320394307587049540576508167804540175310132100\\ 19210293203943075870498404854057650816780454012346100\\ 71970128910238203548304694067960457640175101532100\\ 6196012891023782034730459807640595046540175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023802055830469540679604576401741014532102\\ 71296707173103780384504965406590594765402743201672100\\ 71296701731037803845048765505456901231013456100\\ 71970128102382035498304694067960457640175310132100\\ 189210293203943075870496740576508167802345201256100\\ 7197012801023802055498304694067760457640175310132100\\ 1892102932039430758704985405765091678901231013456100\\ 7197012802035493047540497405765091678901231013456100\\ 71970128010238203549830459405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203493047540487405765091678901231013456100\\ 719701289020394307576494555555555555555555555555555555555$		671602226016 671604206162 671604207013 671604207013 6716042207013 6716042207013 67160422007 67160422007 6716050042017 705141155501 720261526700 721060371501 721240364522 721240364522 721240565203 72130145303 72130145303 72130145303 721443024703 725250020423 7415022015114 74150244163 74150220145 745543010213 750725100216 751522120145 75152210423
$\begin{array}{c} 719701291023802035483046984067960457640175310132100\\ 718970128102382035483046984067960457640175310132100\\ 71897012810238203548304698540679604576401745310132100\\ 1921029320394307587049540576508167804540175310132100\\ 19210293203943075870498404854057650816780454012346100\\ 71970128910238203548304694067960457640175101532100\\ 6196012891023782034730459807640595046540175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701289102382035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023892035483046954067960457640175101532100\\ 719701291023802055830469540679604576401741014532102\\ 71296707173103780384504965406590594765402743201672100\\ 71296701731037803845048765505456901231013456100\\ 71970128102382035498304694067960457640175310132100\\ 189210293203943075870496740576508167802345201256100\\ 7197012801023802055498304694067760457640175310132100\\ 1892102932039430758704985405765091678901231013456100\\ 7197012802035493047540497405765091678901231013456100\\ 71970128010238203549830459405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203543047540487405765091678901231013456100\\ 7197012890203493047540487405765091678901231013456100\\ 719701289020394307576494555555555555555555555555555555555$		671602226016 671604206162 671604207013 671604207013 6716042207013 6716042207013 67160422007 671605006216 671605006216 671605004017 705141155501 720261526700 721060371501 72141222701 721240364522 721240565103 72130145303 72141222501 721442222714 72144222114 721442231303 72144222114 721442231303 72144222114 721442231303 72144222114 741501424703 745542120145 745542120145 745542120145 745542120145 745542120145 745542120145 745542120145 751522120423 75152210423 75152210423 75152210423 75152210423 75152210423 75152200223 75152200223 75152210423 75152210423 75152200223 75152200223 75152200115
719701291023802035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         718970128102382035483046984067960457640175310132100         193704239404954058650598506876091789024562013227100         189210293203943075987049540576598106786091789024562013227100         189210293203943075987049540576509167890123456100         7197012891023820354983046940679604576509167890123456100         719701289102382035498304694067960457640175101532100         6196012891023782034730499840657960476017410145321200         71970129102389203548304694067960457640175101532100         719701291023892035483046984067960457640175101532100         719701291023892035483046984067960447601741014532100         719701291023892035483046984067960457640175101532100         7197012910238920354830469840679604576401741014532100         71970129102389203548304698540679604576509147890123101455200         7197012810238203498304594056765091678091231013456100         718970128102382035498304696556755091678091231013456100         71897012810238203549830469645765091678001231013456100         719210238203943047540497405765091678091231013456100         719701289102382035830469645765091678001231013456100         71970128910238203583046964576509167801231013456100         71970128910238203583046960776509167801231013456100         719701289102382035830469607765091678091231013456100		671602226016 671604206162 671604207015 671604207015 671604227016 6716042207015 671604242063 671604242063 671605004023 671605004023 671605004023 720261526700 720261526700 721060371501 72141155501 72140565203 721240565203 721240565103 72140565203 721442221714 721442231303 72144222714 721442231303 721442221714 721442231303 721442024703 7215256020423 741502244523 741502244533 74150224523 74150224523 74150224523 74150224103 745542120145 755537000423 750725100216 751522120423 751522102423

-- 50 ---

718970128102398203549304694067960457404764017532100	0	0	751526001017
192102983203843048754057650916789012310345301356100	õ	ĩ	760262332041
719701291023920354893046840679860457404764017532100	ō	o i	760473404411
129101971023489204584056798501651021532035430176100	ō	i	760473440023
396302159320236720274201478510589506976079870124100	ō	i	760661510211
192102932039430478404854049740586509168790123456100	ō	ò	760665100415
956790219202972027682028320386543045940149101234100	ŏ	õ	760665400115
718970128103983023820935490469406796045764017532100	ŏ	ĩ	761060547022
189101921029320398430758704854057650816780123456100	ŏ	ò	761060564212
719701289102382049840354830946906796045764017532100	õ	ĭ	761240370023
	•	•	1012.0010020
928902182029320394307587049854057650167810123456100	0	0	761240626114
619601278910237203473045874056985046540164101432100	ŏ	ĩ	761240626122
192102932039830384307587048540576509167890123456100	ŏ	ò	761240662015
189710812802382034983045940569506796047654017432100	ĭ	ĩ	761241217003
219202932039430498540586506876017891012310134567100	ò	ò	761302416023
712897023482045984067960569501651015321035430176100	ĭ	ĭ	761440613211
719701291023892035483069860468409679045764017532100	°.	i	761442115023
189210293203943075870498404854057650816780123456100	0	0	761442212232
189210293203943047540498404854057650816780123456100	ĭ	õ	7614422222216
192102932039430478540497405865016871091790123456100	1	0	761442222007
172102752057450416540471405665016611091190125456100	,	0	101442225001
192102983203843075870485405765091679078970123456100	0	0	761442407023
718970128102398203593056950546506796047640174532100	ŏ	õ	761442414131
519503293013910245920156101731023784204865401687100	ŏ	ĩ	761442424115
192102983203843058950597504854057650916790123456100	ŏ	0	761462120145
719701291023920354893046840679860457640175310132100	ő	1.	761602414033
192102983203843075870485405765091678903453012356100	ŏ	i	761644201017
719701289102382035498304694067960457404764017532100	ŏ	i	761645000217
928902182029320398430758704854057650167810123456100	ĭ	i	770522502222
192102982028320384307587048540576509167890123456100	i	0	770522540023
192102983203843075870485405765016781091890123456100	o	1	770621310212
17210270320304301301040340310301018181091890123430100	0		110021310212
192102932094590395304865404984068760917890123567100	0	1	770624301043
189210293203983038430758704854057650816780123456100	ĭ	ò	770625100216
97890718701281023982034930158104854051850818180123458100	ò	å	771420341043
192102932039430698604964046540568750917890123457100	ő	ĩ	771422220143
192102932039430758705985049540576509167890123456100	ö	1	771432005043
189710812802398203493045940569506796057650175432100	ŏ	i	771604201017
189710812802398203493045940569506796046540176432100	ŭ	1	771605000217
1891019710812802398203493045940569506796046540176452100	ő	1	775412003045
198101291023920349304594056950679607897021876543200	2	-	777402020423
140101241023420344304344030950619601891021816543200	2		111402020423

### TABLE II

# List of simple imperfect squared squares of orders up to and including 19

C= 1058	5 23*	23*	'RF 23	(12,11)(1,3,7)(11,2)(5)(2,5)(4,1)(3)0
C= 3042 C= 3042 C= 3362	\$ 39* \$ 39* \$ 41*	39*	RF 39 RF 39 RF 41	{20+8+11}(5+3){2+12}(7)(19+8}(5+7){11+2}(9)0 (20+19)(1+3+8+7)(19+2)(5)(2+5)(12+1)(3)(8)0 (23+18)(7+11)(18+3+2)(1+5+3)(4)(2+1)(12)(11)0
C= 4608 C= 4608 C= 4608 C= 5202 C= 5408	S 48* S 48* S 48* S 51* S 52*	48* 48* 48* 51* 52*	RF 48 RF 48 RF 48 RF 51 RF 52	(28,20)(7,5,8)(2,3)(9)(20,8)(1])(12,5)(2,9)(7)0 (28,20)(8,12)(20,9,7)(5,7)(2,5)(1])(3,2)(9)(8)0 (28,20)(1],9)(20,8)(2,7)(8,5)(5,3)(12)(2,9)(7)6 (22,14,15)(13,1)(16)(13,9)(9,4)(4,5)(20)(16,1)(15)0 (28,24)(7,9,8)(24,4)(1,6)(5)(1,7)(4,6)(15)(13)0
C=10890 C= 9248 C= 7688 C= 8192 C= 8450 C= 8450 C= 8978 C= 9248 C= 9248 C= 9248 C= 9248 C= 9522 C= 9800	S 11*** 342*4** S 5 645** S 5 645** S 5 688** S 5 688** S 5 688** S 5 700*	34* 62* 64* 64*	RFF 436 4436 455 664 655 70 70 87 87 87 87 87 87 87 87 87 87 87 87 87	$ \begin{array}{l} (4,3,4) (1,2) (3,2) (1,3) (1,2) (1,2) (4) (1,4) (3) 1 \\ (19,15) (4,5,6) (15,7,1) (5,1) (7) (1,4) (8) (1,6) (5) 0 \\ (33,29) (4,5,20) (29,7,1) (6) (13) (7,13) (9,4) (1,6) (5) 0 \\ (36,28) (9,8,11) (28,8) (3,5) (7,2) (5) (2,9) (7) (20) (16) 0 \\ (36,28) (9,11,8) (28,8) (3,5) (7,2) (5,9,2) (7) (20) (16) 0 \\ (36,29) (1,3,8,20) (32,2) (5) (13) (2,5,13) (12,1) (3) (8) 0 \\ (36,29) (16,13) (14,13,9) (4,9) (4,20,1) (5) (1,16) (15) (14) 0 \\ (39,28) (12,7,9) (5,2) (11) (28,8,3) (2,7,80) (5) (12) (15) (14) 0 \\ (25,20,23) (5,12,3) (26) (23,7) (19) (20,3) (7,19) (17,5) (12) 0 \\ (36,29) (18,9,15) (32,4) (11,4) (10) (4,11) (17,8) (1,10) (9) 0 \\ (39,30) (7,12,11) (2,5) (30,11) (3,8) (8,7,2) (5) (20) (19) 0 \\ (39,30) (15,7,19) (32,8) (4,11,8) (5) (7,116) (4,21) (17) 0 \\ (39,30) (5,7,19) (32,2) (1,8) (33,12) (20,3) (5,1) (20) (19) 0 \\ (39,30) (5,7,19) (32,2) (1,8) (33,12) (2,5) (32,1) (20) (13) 10 \\ \end{array}$

C=15488	S 44* 44*	RF176	(24,20)(8,12)(20,4)(7,5)(4,3,5)(2,3)(9)(1,2)(8)(7)0
C=12800	S 80* 80*	RF 80	(44,36)(8,28)(36,16)(9,7)(5,7,16)(2,5)(11)(3,2)(9)(8)0
C=12800	5 80* 80*	RF 80	
C=12800	5 80* 80*	RF 80	(44,36)(11,9,16)(36,8)(2,7)(8,5)(5,3)(28)(2,9)(7)(16)0
			(44,36)(16,7,5,8)(2,3)(9)(36,8)(11)(28,5)(2,9)(7)(16)0
C=12800		RF 80	(51,29)(14,15)(13,1)(16)(29,13,9)(9,4)(4,5)(20)(16,1)(15)0
C=13448	S 82* 82*	RF 82	(47,35)(12,23)(35,12,5,7)(3,2)(1,5,3)(4)(2,1)(24)(23)0
C=13448	S 82* 82*	RF 82	(51,31)(15,16)(9,5,1)(4,13)(31,16,4)(9)(13)(22)(15,1)(14)0
C=14112	5 84* 84*	RF 84-	(44,40)(7,13,20)(40,4)(1,6)(5)(11,6,7)(5,1)(4,24)(20)0
C=14112	S 84* 84*	RF 84	(48+36){9+7+20}(2+5){8+3}(36+7+5)[8)(2+11)[9)(28)(20)0
C=14450	S 85* 85*	RF 85	(39+25+21)(4+17)(15+14)(1+16)(21+17+1)(15)(16)(31)(4+29)(25)0
C=14450	S 85* 85*	RF 85	(46,39)(9,11,19)(39,7)(7,2)(5,8)(5,2)(3,11)(8)(27)(19)0
C=14792	S 86* 86*	RF 86	(47,39)(7,5,8,19)(2,3)(9)(39,8)(11)(12,5)(2,28)(7)(19)0
C=14792	S 86* 86*	RF 86	(47,39)(8,12,19)(39,9,7)(5,7)(2,5)(11)(3,2)(28)(8)(19)0
C=14792	S 86* 86*	RF 86	(47,39)(11,8,20)(39,8)(3,5)(5,7,2)(7)(11,2)(9)(27)(20)0
C=14792	5 86* 86*	RF 86	
C=15138	5 87* 87*	RF 87	(51,35)(15,20)(1,5,9)(35,13,4)(9)(4,16)(13)(22)(1,15)(14)0
C=15138	S 87* 87*	RF 87	(48,39)(7,12,20)(2,5)(39,11)(9,8)(8,3)(28)(2,7)(5)(20)0
	5 0/4 0/4		(48,39)(8,11,20)(3,5)(39,7,2)(5)(2,9)(7)(12)(8,28)(20)0
C=15138	5 87* 87*	RF 87	(51,36)(14,22)(1,13)(36,16)(9,13)(4,9)(20)(5,4)(1,16)(15)0
C=16562	5 91* 91*	RF 91	(52,39)(15,24)(39,7,6)(4,6,5)(1,9)(8)(1,4)(7)(28)(24)0
C=29160	S 18* 18*	RF810	(11,7)(3,4)(1,2)(7,3,2)(1,3)(1,2)(1,2)(4)(1,4)(3)0
C=28800	5 30* 30*	RF480	(13,8,9)(7,1)(10)(8,3,2)(1,5,3)(4)(2,1)(11)(1,10)(9)0
		RF220	(28+27) (7+8+12) (16+12) (6+1) (5+4) (16) (8+15) (11+5) (1+7) (6)0
C=24200			
C=25088	S 56* 56*	RF224	(24,14,18)(10,4)(7,15)(14,10,9,1)(8)(1,8,23)(4,7)(18)(15)0
C=25088	S 56* 56*	RF224	(30+12+14)(7+3+2)(1+15)(4)(11)(4+11)(26+12+3)(7)(2+16)(14)0
C=25992	S 57* 57*	RF228	(30,27)(11,16)(11,7,12)(4,3)(4,7)(1,2)(16)(15,3)(2,14)(12)0
C=26912	S 58* 58*	RF232	(30,28)(2,6,7,13)(28,4)(9,1)(8)(4,9)(15,2)(6)(1,8)(7)0
C=28800	S 60* 60*	RF240	(32,28)(4,5,6,13)(28,7,1)(5,1)(7)(1,4)(8)(1,19)(5)(13)0
C≖30752	S 62* 62*	RF248	(33,29)(8,9,12)(29,4)(11,1)(7,3)(4,11)(3,8)(14)(1,10)(9)1
C=30752	S 62* 62*	RF248	(34,28)(5,8,15)(1,4)(28,7)(1,7)(5)(6,1)(5,1)(4,19)(15)0
C=27556	S 83* 83*	RF166	(31,22,30)(9,13)(7,23)(23,17)(11,2)(9)(13,7)(6,11)(30)(29)(24)0
C=22472	S 106* 106*	RF106	(55,51)(4,5,11,31)(51,7,1)(6)(24)(7,24)(20,4)(1,6)(5)(11)0
C=22472	S 106* 106*	RF106	(55,51)(4,11,36)(51,8)(1,10)(9)(19)(8,9,19)(15,4)(11,1)(10)0
C=22472	S 106* 106*	RF106	(59,47)(12,35)(47,24)(2,5,4,24)(23,1)(3)(1,3)(7,2)(5)(12)0
C=22472	5 106* 106*	RF106	(59,47) (24,23) (47,12) (2,5,4,12) (35,1) (3) (1,3) (7,2) (5) (24)0
	S 107* 107*		(54,53)(1,3,8,21,20)(53,2)(5)(13)(2,5,13)(33,1)(3)(8)(21)0
C=22898		RF107	
C=22898	S 107* 107*	RF107	(59,48)(11,13,24)(48,8,10,4)(2,11)(6)(6,2)(4,14)(35)(10)(24)0
C=22898	S 107* 107*	RF107	(59,48)(11,13,24)(48,14,6,2)(4,11)(10)(35)(10,4)(2,8)(6)(24)0
C=23328	S 108* 108*	RF108	(64,44)(16,11,17)(5,6)(20,1)(24)(44,20)(4,11,5)(24)(6,23)(17)0
C=23328	S 108* 108*	RF108	(64,44)(20,24)(44,24,11,5)(1,23)(6)(17)(20,4)(6,17)(16,5)(11)0
C=23762	S 109* 109*	RF109	(58,51)(7,10,11,23)(51,11,3)(12,1)(12)(7,4)(3,13)(35)(10)(23)0
C=24200	S 110* 110*	RF110	(57,21,32)(10,11)(9,1)(8,36)(17)(53,21)(10,9,17)(1,8)(32)(25)0
C=24200	S 110* 110*	RF110	(57,24,29)(13,6,5)(1,33)(7)(20)(53,24)(13,20)(6,7)(29,1)(28)0
C=24200	S 110* 110*	RF110	(57,53)(4,5,24,20)(53,7,1)(6)(13)(7,13)(33,4)(1,6)(5)(24)0
C=24200	S 110* 110*	RF110	(57,53)(8,9,36)(53,4)(11,1)(10)(21)(4,11,21)(17,8)(1,10)(9)0
C=24200	S 110* 110*	RF110	(60,27,23)(4,19)(16,15)(1,33)(17)(50,27)(16,17)(23,4)(19,1)(18)0
C=24642	5 111* 111*	RF111	(60,51)(9,8,15,19)(1,7)(51,19)(13,9)(6,13)(8,1)(7)(32)(28)0
C=24642	S 111* 111*	RF111	(60,51)(11,15,25)(51,9)(7,4)(19)(16)(9,16)(26,11,7)(4,19)(15)0
			(62,49)(20,29)(49,6,7)(5,1)(4,13,6,5)(9)(1,4)(7)(33)(29)0
C≈24642	S 111* 111* S 111* 111*	RF111 RF111	(68,43)(23,20)(3,17)(7,19)(43,20,5)(12)(5,12)(3,26,7)(23)(19)0
C=24642			
C=25088	S 112* 112*	RF112	(60,52)(8,13,15,16)(52,16)(11,2)(9,8)(7,9)(13,2)(36)(11)(24)0
C=25088	S 112* 112*	RF112	(64+48)(7+5+8+28)(2+3)(9)(11)(48+20+5)(2+9)(7)(8+36)(28)0
C=2508B	5 112* 112*	RF112	(64,48)(20,28)(48,9,7)(5,7,8)(2,5)(11)(3,2)(9)(8)(36)(28)0
C=25538	5 113* 113*	RF113	(66,47)(23,24)(47,19)(5,7,1)(6,19)(13)(28,6)(1,8,23)(7)(15)0
C=25538	5 113* 113*	RF113	(68,45)(26,19)(7,12)(45,20,3)(12,19,5)(17)(5,7)(25)(23,3)(20)0
C=25992	S 114* 114*	RF114	(41,39,34)(5,29)(2,11,31)(34,9)(20)(9,20)(5,44,2)(11)(39)(31)0
C=25992	5 114* 114*	RF114	(60,29,25)(4,21)(16,17)(15,1)(39)(54,21)(17,4)(14,29)(16,1)(15)0
C=25992	S 114* 114*	RF114	(63,51)(12,11,28)(1,10)(51,13,12)(3,7)(4,11)(10,3)(7)(35)(28)0
C=26450	S 115* 115*	RF115	(65,50)(16,20,14)(5,9)(50,14,1)(13,4)(1,4)(9,16)(13)(36)(29)0
C=26450	S 115* 115*	RF115	(67,48)(11,9,28)(2,7)(8,5)(12)(48,20,7)(5,2)(3,39)(8)(28)0
C=26450	S 115* 115*	RF115	(68,47)(15,32)(6,9)(47,15,8,4)(1,8)(5)(7,1)(6)(4,36)(32)0
C=26912	5 116* 116*	RF116	(60,56)(4,12,6,7,27)(56,8)(5,1)(8)(1,4)(21)(12)(6,33)(27)0
C=26912	5 116* 116*	RF116	(60,56)(4,12,13,27)(56,8)(19,1)(14)(8,33)(15,4)(12)(14,1)(13)0
C=26912	S 116* 116*	RF116	(60,56)(4,12,15,11,14)(56,8)(8,3)(17,3)(17)(14,4)(12)(31)(29)0
C=26912	S 116* 116*	RF116	(60,56)(4,12,15,25)(56,8)(10,7,3)(8,10)(3,4)(13)(12)(35)(25)0
C=26912	S 116* 116*	RF116	(65,51)(9,13,29)(5,4)(1,16)(51,20)(9,36)(16,4)(13)(15,1)(14)0
C=26912	S 116* 116*	RF116	(68,48)(23,25)(48,17,3)(19,7)(5,20)(12)(12,5)(7,26,3)(23)(19)0
	S 117* 117*		(65,52)(8,12,32)(5,3)(2,1)(13)(52,20)(8,5)(3,2)(1,33)(32)0
C=27378 C=27378	5 117* 117*	RF117 RF117	(68,49)(26,23)(49,12,7)(3,20)(5,19,12)(17)(7,5)(25)(3,23)(20)0
C=27378	S 117* 117* S 118* 118*	RF117 RF118	(70,47)(18,29)(3,4,11)(2,1)(5)(47,18,7)(3,2)(1,41)(11)(29)0 (62-56)(13-8-7,20)(56-6)(1-6)(6-5)(3)-1)(12)(8-6)(32)(29)0
C=27848			(62,56)(12,8,7,29)(56,6)(1,6)(4,5)(21,1)(12)(8,4)(33)(29)0
C=27848	5 118* 118*	RF118	(69,49)(8,11,30)(5,3)(2,12)(7)(49,19,8)(5,7)(11,2)(39)(30)0
C=28322	S 119* 119*	RF119	(70,49)(17,32)(4,6,7)(49,17,8)(5,1)(8)(1,4)(9)(6,38)(32)0
C=28800	S 120* 120*	RF120	(62,58)(8,9,15,26)(58,4)(11,1)(10)(4,11)(17,8)(1,36)(9)(26)0
C=28800	5 120* 120*	RF120	(70,50)(19,31)(1,3,8,7)(50,19,2)(5)(2,5)(12,1)(3)(39)(31)0
C=29768	S 122* 122*	RF122	(52+33+37)(29+4)(41)(29+13+10)(3+23+13)(16)(10+3)(44)(41+4)(37)0
			(66,56)(12,13,31)(56,10)(8,4)(3,10)(7)(15,3)(12,8)(4,35)(31)0
C=29768	5 122* 122*	RF122	(00,50)(12,15,51)(50,10)(8,4)(5,10)(7)(12,5)(12,0)(4,55)(51,0)
C=29768 C=31752	S 122* 122* S 126* 126*	RF122 RF126	(68,58)(11,15,32)(58,9,1)(8,4)(10,9)(17)(1,8)(11)(4,36)(32)0

# 

# TABLE III

# A few examples of squarings obtained from various nets

<b>6-</b>	130 S	i5*	11≉	RF ·	5 (6,4,5)(3,1)(6)(5,1)(4) 100
Ċ-	130 S	69 <b>*</b>	61		(36,33)(5,28)(25,9,2)(7)(16) 1 0 0
<u>د</u> ه د	130 S 130 S	53* 33*	52 32	EF	2(18,15)(7,8)(14,4)(10,1)(9) 1 0 0 2(18,15)(7,8)(14,4)(10,1)(9) 1 0 0
с.	130 8	33×			2 (18,15)(7,6)(14,4)(10,1)(9) 1 0 0
C=	.130 S	69 <b>*</b>	61		(36,33)(5,28)(25,9,2)(7)(16) 1 0 0
0- 0-	130 S	15* 33*	11;* · 70	RF	5 (6,4,5)(3,1)(6)(5,1)(4) 1 0 0 2 (16,15)(7,8)(14,4)(10,1)(9) 1 0 0
č.,	130 \$		61	r.r	(36,33)(5,28)(25,9,2)(7)(16) 1 0 0
0.	130 S	69 <b>*</b>	61		(36,35)(5,28)(25,9,2)(7)(16) 1 0 0
	31752 S 31752 S				24 (357,307)(47,107,153)(3,44)(302,58)(17,11,16)(6,5)(82,46)(81)(199)(163) 000 3 (2480,1408,1717)(538,501,369)(596,1550)(37,464)(575)(148,912)(2439,41)(764)(1634,42)(1592) 000
	31752 S		4847	RF	3 (2552,1408,1777)(1039,369)(842,1304)(143,724,172)(2295,219,38)(181)(552,462)(400)(1766)(1676) 0 0 0
	31752 s		2426	$\mathbb{RP}$	6 (1218,744,904)(484,260)(100,165,639)(295,65)(230)(1208,10)(423,71)(352,244)(883)(775) 0 0 0
	31752 S 31752 S		585*	RF	27 (320,271)(41,94,136)(8,33)(265,41,22)(19,3)(16,20)(76)(72,42)(178)(148) 000 24 (360,310)(46,68,176)(4,42)(293,71)(33,97)(104)(9,167)(40,66)(118,26)(92) 000
	31752 6				27 (304, 150, 163)(137, 13)(176)(25, 60, 52)(255, 41, 8)(33)(8, 220)(72, 8)(70)(142) 0 0
	31752 S		661	RF :	24 (342,156,164)(148,6)(172)(38,78,32)(46,158)(319,61)(21,103)(82)(176,9)(167) 0 0 0
	31752 B 31752 B		4729	ICF D	3 (2385,1570,1900){856,714){364,1516}(142,956){2344,41}(1039){225,555,176}(379,1513){1864}(954) 0 0 0 3 (2390,1376,944,1313){575,369}(334,1346){1233,143}(590,188){462}(500,552){2171,219}(1952)(1900) 0 0 0
					6 (1411,1275)(136,464,675)(1195,352)(24,210,230)(373)(190,20)(19,656)(269)(467,99)(368) 0.0 0
Ç⇒	31752 8	5591*	4993	RF	3 (2822,2769)(117,428,2224)(2171,379,208,64)(181)(171,37)(88,130)(46,42)(600)(596)(1196) 0 0 0
	31752 S 31752 S				9 (923,850)(73,185,176,416)(832,164)(9,167)(52,112)(216)(126,16)(110,73)(489)(452) 0 0 0 24 (312,198,223)(118,50)(25,42,156)(58,17)(59)(40,18)(278,34)(22,55)(244)(211) 0 0 0
	31752 S				24 (512,196,225)(146,50)(25,42,156)(56,17)(59)(40,16)(276,54)(22,55)(244)(211) 0 0 0 12 (495,419,452)(76,194,149)(116,356)(339,232)(45,220)(239)(107,125)(64,492)(446)(428) 0 0 0
	31752 S	1450*	1196	RF	12 (600,428,422)(123,299)(130,181,117)(64,176)(42,88)(596,46)(37,208)(171)(475)(379) 0 0 0
	31752 S 31752 S				6 (1228,817,801)(16,785)(421,412)(1218,10)(9,213,190)(440)(23,167)(236)(92,860)(768) 000 12 (642,336,430)(242,94)(191,335)(64,135,43)(92,142)(596,110)(39,188)(149)(475)(337) 000
	31752 S				12 (0982, 555, 951/242, 954, 191, 555) (091, 155, 53) (52, 142) (595, 110) (59, 165) (149) (415) (57) (50 (50) 12 (107, 64, 53) (1, 62) (27, 36) (16, 11) (26, 23) (100, 23) (3, 82) (2, 27) (25) (52) 0 0 0
0=	31752 8	126*			26 (58,56)(11,15,32)(58,9,1)(8,4)(10,9)(17)(1,8)(11)(4,36)(32) 0 0 0
	52893 s				(11464,7398,11407)(4066,5332)(1111,1544,677)(867,11217)(11160,2868,1102)(725,396)(339,47)(2458)(2166)(7892) 000
	52893 S 52893 S			FF	9 (1322,966,1240)(312,360,274)(405,1109)(44,200,68)(1027,339)(149,299)(183,17)(166)(704)(688) 0 0 0
	52893 s			RP	(12289,7530,8887)(\r59,1739,1032)(707,325)(2828,6364)(2446)(1718,3556)(11898,4006,1144)(2862)(1024,5916)(7892) 0 0 0 3 (4283,2551,2775)(1435,592,224)(724,2275)(836,56)(780)(297,1138)(845,771)(3739,841)(74,2972)(2898) 0 0 0
C= (	52893 s	31361*	21532		(12289,5491,10581)(3576,3440,1475)(860,615)(245,10951)(1105)(136,4409)(222,3490)(9243,3268)(5975,783)(5192) 0 0 0
	52893 S 52893 S			RF	3 (4756,2201,2391)(2011,190)(2581)(1461,790,760)(83,378)(3527,1017,295)(30,3311)(49,771)(722)(2510) 0 0 0 (9933,8491,8887)(1442,3799,2854,396)(2458,6885)(6756,4619)(945,456?)(4744)(2137,2482)(1322,9870)(8893)(8548) 0 0 0
	52893 s				(12849,7458,8330)(4367,2219,872)(1565,7637)(2001,218)(1783)(147,3637)(1024,3490)(11407,2466)(8941,652)(8289) 0 0 0
C=	52893 É	990*	969	RF 2	7 (560,430)(74,124,232)(43,31)(12,19)(13,42)(35,108)(409,135,29)(106)(33,307)(274) 000
	52893 s 52893 s				(11464,8941,11043)(2523,3675,2743)(641,10402)(9981,4006)(932,8452)(8854,1753)(233,2219)(1986)(5975,685)(5090) 0 0 0 (15120,12251)(3521,4280,4450)(10402,4066,652)(3414,759)(3799,1240)(1070,3380)(2310)(6336,1144)(249,5441)(5192) 0 0 0
	52893 S				(14268,6558,6848)(6268,290)(7138)(1442,3666,1160)(2506,5792)(10951,4759)(2535,3637)(6192,1102)(351,5441)(5090) 0 0 0
	52893 s				(14110,12666)(1444,4619,6603)(12007,3547)(372,2508,1739)(3919)(1494,245)(6848)(1783,725)(2219)(4541,1161)(3380) 0 0 0
	52893 S			RF	3 (4610,4222)(388,878,770,2186)(4189,809)(108,662)(319,667)(1128)(113,549)(780)(344,205)(2391)(2252) 0 0 0 (12251,5792,4541,6448)(2634,1907)(4409,1383)(727,7528)(4744)(2691,1718)(4305,2157)(11610,641)(3352)(3785)(7637) 0 0 0
C=	52893 Б	27940*	24953		(14110,13830)(570,2137,11123)(10843,1907,1070,290)(860)(837,233)(153,707)(386)(218,168)(3012)(2962)(5974) 000
	52893 3 52893 S				(12567,77135,6893)(6268,1473)(325,2172,6396)(1800)(1753,47)(2219)(11123,1444)(1240,513)(727,2005)(9679)(8401) 0 0 0 (12007,7138,1450,6448)(2452,1998)(1565,6881)(1341,1111)(6033,1105)(2676)(2446)(2482,2640)(10843,1164)(9679)(9521) 0 0 0
	52893 S				(120k8,85k8,8401)(2k27,557k)(2828,3440,2280)(1160,3547)(672,1544,612)(322,4280)(118k8,872)(3343)(9521)(7628) 0 0 0
C=	52893 8	9438*	8195	PP	3 (\$177,2\$26,2775)(972,1225,289)(936,2125)(719,253)(1222,1192)(\$016,880)(30,3290)(124,1128)(1004)(2132) 0 0 0
C=	52893 s	28514*	24379		(1253),7653,8330)(3556,3420,677)(2743,6254)(136,2506,3521)(1322,2370)(11848,2005)(957,3919)(2562)(9785)(6881) 0 0 0
~	59508 s	16/100	الم <sup>ا</sup> لم	DP -	8 (805 755)(152 725 380)(70) 105 20)(10) 70)(57 3)(5) 15)(107 24)(15)(3) 34)(16)(3) 34)(16)
	59508 S	891*			8 (896,766)(152,225,389)(748,125,22)(104,70)(67,3)(64,164)(197,33)(164)(64,489)(425) 000 6 (383,239,269)(148,91)(61,208)(61,91)(379,4)(148,4)(65)(35,56)(79,21)(285)(227) 000
	59508 s		1652*	RF 1	8 (896,758)(149,191,418)(756,129,11)(112,42)(118,115)(205,42)(3,112)(163)(54,476)(422) 000
	59508 S 59508 S				8 (748,501,565)(197,240,64)(210,389)(50,104,13)(213,64)(744,54)(158)(155,149)(558)(552) 0 0 0 8 (794,498,476)(22,454)(243,277)(55,156,34)(156,155)(744,103)(225,34)(1,608)(191)(416) 0 0 0
	59508 S				4 (16,14,15)(2,7,5)(4,11)(11,7)(2,7)(2,7)(4,5)(2,16)(15)(14) 0 0 0
	59508 s		752*	RF 🕽	6 (378,249,274)(133,91,25)(98,201)(59,32)(27,103)(374,4)(120,17)(103)(27,277)(250) 000
					8 (785,425,554)(296,783)(245,440)(64,156,76)(757,59)(122,197)(205,42)(154)(69,548)(459) 0 0 0 8 (785,500,489)(11,478)(234,277)(51,140,43)(130,190)(747,69)(196,35)(155)(103,565)(462) 0 0 0
C=	59508 s	1802*	1504	RF 1	8 (757, 497, 548) (206, 240, 51) (197, 402) (54, 113, 34) (266, 6) (205) (747, 64) (182) (55, 554) (501) 0 0
	59508 s		77)*	RF ;	6 (397,211,274)(148,63)(117,220)(38,78,32)(44,103)(374,61)(21,103)(82)(46,277)(231) 000
	59508 s 59508 s				6 (378,249,274)(133,91,25)(98,201)(59,32)(27,103)(374,4)(120,17)(103)(27,277)(250) 000 8 (896,758)(149,191,418)(756,129,11)(118,42)(118,115)(205,42)(3,112)(163)(54,476)(422),000
¢≃	59508 s	1802*	1504	RF 1	8 (757,497,548)(206,240,51)(197,402)(54,118,34)(265,8)(205)(747,64)(182)(53,554)(501) 0 0
	59508 s		1542	RF 1	8 (785,425,554)(296,129)(243,440)(64,156,76)(757,92)(122,197)(206,42)(164)(89,548)(459) 000
	59508 S 59508 S		762#	RF 1	8 (896,756)(152,225,289)(748,125,22)(104,70)(67,3)(64,164)(197,33)(164)(64,489)(425) 000 6 (383,239,269)(148,91)(61,206)(61,91)(379,4)(148,4)(65)(35,56)(79,21)(285)(227) 000
<b>c</b> = ;	59508 s	1774*	1532	RF 1	8 (785,500,489)(11,478)(234,277)(51,140,43)(130,190)(747,89)(196,33)(163)(103,565)(462) 0 0 0
<b>0</b> -	59508 a	1814*	1492*	R9" 1	8 (748,501,565)(197,240,64)(210,529)(50,104,43)(219,64)(744,54)(158)(155,149)(538)(532) 000
с. С	59508 8 59508 8	1768# 882#			8 (754,498,476)(22,454)(243,277)(53,156,34)(156,155)(744,103)(225,34)(1,608)(191)(416) 000 6 (397,211,274)(148,63)(117,220)(38,78,32)(46,103)(374,61)(21,103)(82)(46,277)(231) 000
-					

C= 9	99072 5	19023*	14001	RF	3 (7071,5000,6952)(2071,1481,1448)(33,911,504)(916,598)(407,7049)(6930,1886,526)(318,280)(1598)(1560)(5044) 0 4
<b>C=</b> 9	<i>79072</i> s	3306*	2198×	$\overline{kr}$	18 (1244,907,1155)(299,360,248)(360,1043)(38,200,61)(954,328)(173,243)(166,34)(132,75)(685)(626) 0 0 0
0-9	99072 s	4508*	3748*	RF	12 (1882,1224,1402)(658,380,186)(8,429,965)(194)(347,227)(120,536)(1866,605,69)(536)(120,1381)(1261) 0 0 0
0- S	99072 s	18015*	15009	RF	3 (7960,4808,5247)(2456,1913,439)(1448,4238)(26,1422)(543,1396)(696,2303)(7049,1607)(1450,1368)(82,5524)(5442)
C= 9	99072 s	9866+	6626*	RF	6 (3764,2690,3432)(1033,1041,616)(490,126)(364,3194)(41,984,8)(984,65)(2862,943)(919)(1919,8)(1911) 0 0 0
0= 5	99072 s	744×	632*	RF	72 (346,204,194)(10,184)(103,111)(39,56,8)(56,63)(286,82,17)(65,8)(57,7)(254)(204) 000
C= 5	99072 s	8542*	7970*	RF	6 (3048,2690,2804)(358,1305,913,114)(799,2119)(2082,1324)(392,1320)(377,1320)(758,943)(392,3047)(2840)(2655) 0
c= s	99072 s	2264*	1864*	P.P	24 (995,665,664)(330,216,59)(235,488)(138,78)(30,253)(60,246,24)(222)(869,186)(29,712)(685) 0 0 0
C= S	99072 s	2232*	1896*	RF	24 (1001,567,664)(268,202,97)(241,520)(66,136)(166,168)(98,279)(395,270,2)(268)(87,712)(625) 0 ^ 0
C= 9	99072 s	19695*	13329	RP	3 (7160,5464,7071)(1656,2201,1607)(2420,6258)(40,1071,545)(6169,1031)(920,1826)(1708,394)(1314)(4-8,3833)(3430)
0+ 9	99072 s	18847*	14177	R7	3 (8008,5272,5567)(2504,2473,295)(2010,5852)(168,1842)(31,2610)(232,2303)(61±9,2071)(936,4758)(4098,276)(3822)
0- 5	99072 s	9166*	7346*	RP.	6 (4152,2202,8812)(1592,610)(1267,2155)(358,249,285)(664,888)(3194,1316)(725,888)(664,2379)(1878,165)(1715) 0 4
C= 5	99072 s	4508*	3748*	RF	12 (1878,1224,1406)(662,580,182)(198,427,963)(349,229)(120,536)(1870,8)(603,67)(536)(120,1379)(1259) 0 0 0
C= 9	99072 s	2762*	2742*	$\mathbb{RP}$	18 (1490,1272)(245,333,694)(1252,211,27)(184,68)(184,237)(307,88)(219,53)(166,124)(42,776)(734) 0 0 0
0= 5	99072 s	2264*	1864*	RF	24 (941,659,664)(222,246,186,5)(181,488)(60,307)(60,138,24)(330)(923,78)(216)(83,712)(629) 0 0 0
Ce S	99072 s	17631*	15393	$\mathbf{F} \boldsymbol{F}$	3 (7865,4250,5516)(2954,1266)(2472,4310)(631,1593,754)(7528,968)(1388,1838)(2024,543)(1481,450)(1031,5567)(4536
C# 5	99072 s	8302*	8210*	FF	6 (4470,5832)(758,1133,1941)(3740,610,120)(490,388)(13,312,606)(102,299)(1005,197)(608)(512,2437)(2125) 0 0 0
c= 9	99072 s	2216*	1912*	RF	24 (958,589,669)(373,216)(136,553)(136,216)(21,115)(954,4)(398)(35,181)(150)(4,710)(154)(552) 0 0 0
C≥ 9	99072 s	17775*	15249	fø	3 (7865,5036,4874)(162,4\$12)(2534,2664)(295,1071,1033,130)(908,1886)(7384,776)(35,1913)(1680)(935,5663)(4728)
0= 9	99072 S	18127*	14897	$\mathbb{RP}$	3 (7465,5310,5352)(1716,2082,1512)(1470,5882)(570,2412)(439,911,366)(545,2473)(7432,472)(1988)(631,5663)(3032)
C= 9	99072 s	17823*	15201	RF	3 (7736,4840,5247)(2232,2201,407)(1794,3860)(51,1898,2066)(664,1509)(7465,935)(2266,268)(1998,168)(6094)(4264)
0= 5	99072 s	2232*	1896*	RF	24 (967,601,664)(268,270,65)(207,520)(98,168,2)(166,313)(229,136)(66,268)(202)(121,712)(591) 0 0 0

### PROGRAMME I

```
begin integer N, x : N := 8;
  begin integer number of choices:
       Boolean selfdual:
       integer array W[1:2*(2*N \div 3 + N) + 1];
       procedure wheel (B); value B;
                           integer B;
       begin integer MDP, t, l;
            MDP: = B \div 2 + 1: t: = 1:
            for l := 1 step 1 until MDP-2 do
                begin
                  W[t]: = W[t+3]: = l; W[t+1]: = MDP; W[t+2]: = l+1; W[t+4]: = 0; t = t+5
                end:
             W[t] := W[t+3] := MDP-1; W[t+1] := MDP; W[t+2] := 1; W[t+4] := 0; t := t + 5;
             for l := 1 step 1 until MDP - 1 do
                begin
                  W[t]: = l: t: = t + 1
                end:
             W[t]: = 1; W[t+1]: = W[t+2]: = 0
       end wheel;
       procedure WRITE (W, number of choices, selfdual); integer number of choices;
                                                         Boolean selfdual;
                                                         integer array W;
```

```
begin integer i; write (0); write (W[1]);
```

```
for i := 2 step 1 until i do begin
```

write (W[i]); if  $W[i-1] = 0 \land W[i] = 0$ then go to end

end;

end: write (number of choices);

if selfdual

then write (1)

else write (0)

end WRITE;

write (N); wheel (N); number of choices: = 2; selfdual: = true; WRITE (W, number of choices, selfdual); write (-1); go to finish;

start: begin integer K, M, B, H, end of file, identificationnumber, storage;

then selfdual: = false

integer array branch 1, branch 2, branchdual 1, branchdual 2 [1:N+1], W[1:2\*(2\*N÷3+N)+1], V[1:2\*(2\*N÷3+N)+5];

procedure READ (W, number of choices, selfdual, end of file); integer number of choices, end of file;

**Boolean** selfdual; integer array W;

```
begin integer i, j;

W[1]: = read;

for i: = 1 step 2 until i do

begin

W[i+1]: = read; W[i+2]: = read; if W[i+1] = 0 \land W[i+2] = 0

then go to end

end;

end: number of choices: = read;

j: = read; if j = 0
```

```
else selfdual: = true;
      end of file: = read
end READ;
procedure form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
          integer K, M;
          integer array branch 1, branch 2, branchdual 1, branchdual 2, V;
          begin integer m, t, tt, i;
               t := m := 1; tt := 0;
          begin:
               for i := 1 step 1 until tt do
                   begin
                     if V[t+1] = branch 1 [i] \land V[t] = branch 2[i]
                     then
                       begin
                         branchdual 2[i]:=m; go to next
                       end
                   end i:
               tt: = tt + 1; branch 1[tt]: = V[t]; branch 2[tt]: = V[t+1]; branchdual 1[tt]: = m;
          next: t := t + 1; if V[t+1] = 0
                           then
                             begin
                               if V[t+2] = 0
                                then go to end;
                               m := m + 1; t := t + 2
                             end;
               go to begin;
```

- 57 -

```
end: B := tt: M := m; K := B + 2 - M
          end form branches;
procedure dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, V);
          integer K;
          integer array branch 1, branch 2, branchdual 1, branchdual 2, V;
          begin integer i, j, l, h, t, search, remember;
               integer array vector 1, vector 2[1:B];
               t := 0; i := 1;
          start: l := 1;
               for j := 1 step 1 until B do
                   begin
                     if branch 1[i] = i
                     then
                        begin
                          vector 2[l]: = branchdual 1[j];
                          vector 1[l]: = branchdual 2[j]; l: = l + 1
                        end;
                     if branch 2[i] = i
                     then
                        begin
                          vector 1[i]: = branchdual 1[i];
                          vector 2[l]: = branchdual 2[i]; l: = l + 1
                        end
                   end:
                t: = t + 1; V[t]: = vector 1[1]; search: = remember: = vector 2[1];
         begin: for h := 1 step 1 until l - 1 do
```

```
- 58
```

```
begin
if vector 1[h] = search
then
begin
t: = t + 1; V[t]: = search; search: = vector 2[h];
if search = remember
then go to continue;
go to begin
end
```

end;

continue:

$$t := t + 1; V[t] := 0; i := i + 1;$$
if  $i = K + 1$ 

then go to end;

go to start;

end: t := t + 1; V[t] := 0

end dualize;

procedure identification (V, identificationnumber); integer identificationnumber;

### integer array V;

begin integer i, j, workstorage, maxweight original, maxweight dual, remember maxweight dual, remember max weight original, n original, n dual, l;
Boolean dual with fromdual, fromdual, nogain;
integer array weight original, location original [1:K+1], inverse location [1:K], weight dual, location dual [1:M+1];
procedure identify (K, weight original, weight dual, location, max weight, n, branch 1, branch 2, branchdual 1, branchdual 2);
integer K, n, maxweight;

```
integer array weight original, weight dual, location, branch 1, branch 2, branchdual 1, branch
                    dual 2;
begin integer z, weightstorage, t, i, k, q, s, l, min;
      Boolean ready;
      integer array new location, score [1:K];
٠
      nogain: = false;
start: z: = 1; weightstorage: = 1; t: = 1;
       for i := 1 step 1 until K do score [i] := 0;
       if fromdual
       then for i := 1 step 1 until B do
               begin
                  score [branch 1 [i]]: = score [branch 1 [i]] + weight dual [branchdual 2 [i]];
                  score [branch 2 [i]]: = score [branch 2 [i]] + weight dual [branchdual 1 [i]];
                  fromdual: = false
               end
       else for i := 1 step 1 until B do
               begin
                  score [branch 1 [i]]: = score [branch 1 [i]] + weight original [branch 2 [i]];
                  score [branch 2 [i]]: = score [branch 2 [i]] + weight original [branch 1 [i]]
               end:
label 1:
       for i := z step 1 until K do
           begin
             if weight original [location [i]] \neq weight original [location [i + 1]]
             then go to continue
           end;
```

continue:

.

```
if i > z
then
  begin
    for k := z step 1 until i do weight original [location [k]]: = 0;
    label 2:
    min: = M*2\uparrow K; ready: = true;
    for l := z step 1 until i do
      begin
        if score [location [l]] < \min \land weight original [location [l]] = 0
        then
           begin
             min: = score [location [/]]; ready: = false
           end
      end;
    if ready
    then go to continue i;
    weightstorage: = 2*weightstorage;
    for n := z step 1 until i do
      begin
        if score [location [n]] = min
        then
          begin
             weight original [location [n]]: = weightstorage;
             new location [t]: = location [n]; t: = t + 1
          end
```

- 61

```
end n;
n: = i; go to label 2
end of then i > z
else
begin
weightstorage: = 2*weightstorage;
weight original [location [i]]: = weightstorage;
new location [t]: =location [i]; t: = t + 1
end else;
```

continue *i*:

```
z := i + 1; if z \leq K
                     then go to label 1;
        if weightstorage \neq 2 \uparrow K
        then if weightstorage = maxweight
             then
                begin
                  nogain: = true; go to finish
                end
             else
               begin
                  for s := 1 step 1 until K do location [s] := new location [s];
                  maxweight: = weightstorage; go to start
               end else;
        for s := 1 step 1 until K do location [s] := new location [s];
finish:
end identify;
```

- 62 -

```
number of choices: = 0:
      for i := 1 step 1 until K do
         begin
           weight original [i]: = 2; location original [i]: = i
         end;
      location original [K+1]: = K + 1; weight original [K+1]: = 0; fromdual: = false;
      maxweight original: = 2;
     identify (K, weight original, weight dual, location original, maxweight original, n original, branch 1, branch 2,
              branchdual 1, branchdual 2);
      remember maxweight original: = maxweight original;
     if nogain
      then go to form identificationnumber;
     for i := 1 step 1 until M do
         begin
           weight dual [i]: = 2; location dual [i]: = i
         end:
      location dual [M+1]: = M+1; weight dual [M+1]: = 0; from dual: = false; maxweight dual: = 2;
      dual with fromdual: == false;
      identify (M, weight dual, weight original, location dual, maxweight dual, n dual, branchdual 1, branchdual 2,
              branch 1, branch 2);
      remember maxweight dual: = maxweight dual;
two: fromdual: = true;
three: identify (K, weight original, weight dual, location original, maxweight original, n original, branch 1, branch 2,
              branchdual 1, branchdual 2);
      if ¬ nogain
      then go to form identificationnumber;
```

```
if maxweight original = remember maxweight original
```

then

```
begin
          if dual with fromdual
          then
            begin
            five: weight original [location [n original]]: =
                 weight original [location [n original]] + 1; dual with from dual: = false;
                 number of choices: = number of choices +1; go to three
            end;
          go to four
        end:
      remember maxweight original: = maxweight original;
four: fromdual: = true:
      identify (M, weight dual, weight original, location dual, maxweight dual, n dual, branchdual 1, branchdual 2,
              branch 1, branch 2);
      dual with fromdual: = true;
      if nogain
      then go to two;
      if maxweight dual = remember maxweight dual
      then go to five;
      remember maxweight dual: = maxweight dual; go to two;
form identificationnumber:
      for i := 1 step 1 until K do inverse location [location original [i]]: = i;
      identification number: = 0;
      for l := 1 step 1 until B do
```

```
begin
     i := K + 1—inverse location [branch 1 [/]]; j := K + 1—inverse location [branch 2 [/]];
     if i > j
     then
        begin
          workstorage: = i; i: = j; j: = workstorage
        end;
     identificationnumber := identificationnumber + 2\uparrow((K\uparrow 2+K+i*(i-2*K+1)-2*i)\div 2))
   end
end identification;
procedure form TNSTAR;
begin integer array U[1:2*(2*N+3+N)+5];
     procedure new net test (V, storage); integer storage;
                                          integer array V;
     begin integer p;
           own integer array id number [1:4 \uparrow (B-9)]
           for p := 1 step 1 until H do
               begin
                 if storage = id number [p]
                 then go to end
              end;
           H: = H + 1; id number [H]: = storage; WRITE (V, number of choices, selfdual);
end:
end new net test:
form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
if K = M
```

### then

### begin

```
identification (V, identificationnumber);
storage: = identificationnumber;
dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, U);
form branches (U, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
identification (U, identificationnumber);
if identificationnumber < storage
then
```

### begin

```
selfdual: = false; new net test (U, identificationnumber)
```

# end

### else

else

# begin if identificationnumber > storage then begin selfdual: = false; new net test (V, storage) end else begin selfdual: == true; new net test (V, storage) end end end end se

begin
if K > M
then
begin
dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, U);
form branches (U, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
identification (U, identificationnumber);
selfdual: = false;
new net test (U, identificationnumber)
end
else

begin

```
identification (V, identificationnumber); selfdual: = false; new net test (V, identificationnumber)
end
```

```
end
```

```
end form TNSTAR;
procedure generate nets (W); integer array W;
begin Boolean dualized;
    dualized := false; go to con 2;
con 1: if dualized ∨ selfdual
    then go to finished;
    form branches (W, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
    dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, W);
    dualized := true;
con 2: begin integer i, ii, m, s, t, MM, p, q, a, b, l;
    integer array sum [1:N], multiplicity [1:N];
```

```
m: = t: = \text{sum } [1]: = i: = 1;
label: if W[t+2] = 0
      then
        begin
          t: = t + 3; i: = i + 1; \text{ sum } [i]: = t; \text{ multiplicity } [i-1]: = m; m: = 1;
          if W[t] = 0
          then go to follow
        end:
      t := t + 1; m := m + 1; go to label;
follow:
      MM:=i-1;
      for ii := 1 step 1 until MM do
         begin
            if multiplicity [ii] > 3
            then
              begin
                q := sum [ii] - 1;
                for a := 1 step 1 until sum [ii] - 1 do V[a] := W[a];
                for b: = sum [ii+1] step 1 until sum [MM+1] do V[b+4]: = W[b];
                 for s:=1 step 1 until multiplicity [ii]-2 do
                    begin
                       for l: = s + 2 step 1 until if s = 1 then multiplicity [ii] - 1
                                                          else multiplicity [ii] do
                          begin
                            p:=q+1;
                             for m: = s step 1 until l do
```

- 68 -

```
begin
                                           V[p]: = W[m+q]; p: = p+1
                                         end;
                                      V[p]: = W[s+q]; p: = p+1; V[p]: = 0; p: = p+1;
                                     for m := l step 1 until multiplicity [ii] do
                                         begin
                                           V[p]: = W[m+q]; p: = p+1
                                         end;
                                     for m: = 1 step 1 until s do
                                         begin
                                           V[p]: = W[m+q]; p: = p+1
                                         end;
                                      V[p]: = W[l+q]; V[p+1]: = 0;
                                     form TNSTAR;
                                   end l
                              end s
                       end if
                  end ii
            end block con 2;
            go to con 1;
          finished:
          end generate nets;
     H:=0; write (N+1);
next net:
     READ (W, number of choices, selfdual, end of file);
     generate nets (W);
```

- 69 -

```
if end of file \geq 0

then

go to next net;

N := N + 1; if N - 2*(N \div 2) = 0

then

begin

wheel (N); selfdual: = true; number of choices: = 2; WRITE (W, number of choices, selfdual)

end;

write (-1);

end;

finish: stop; N: = read; x: = read; go to start;

end
```

## **PROGRAMME II**

W[1]: = read;

```
for i := 1 step 2 until i do
         begin
            W[i+1]: = \text{read}; W[i+2]: = \text{read}; \text{ if } W[i+1] = 0 \land W[i+2] = 0
                                                 then go to end
         end:
end: number of choices: = read; j: = read; end of file: = read
end READ;
READ (V, number of choices, end of file);
begin integer K, M, complexity, hcf;
      integer array branch 1, branch 2, branchdual 1, branchdual 2[1:B], INC [1:2*B+3-1, 1:4*B+3-2],
                   ZINV [1:2*B \div 3, 1:2*B \div 3];
      procedure form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
                integer K, M;
                integer array branch 1, branch 2, branchdual 1, branchdual 2, V;
      begin integer m, t, tt, i;
           t:=m:=1; tt:=0;
      begin: for i := 1 step 1 until tt do
                begin
                   if V[t+1] = \text{branch } 1[i] \land V[t] = \text{branch } 2[i]
                   then
                       begin
                          branchdual 2[i]: = m; go to next
                       end
                end i:
             tt: = tt + 1; branch 1[tt]: = V[t]; branch 2[tt]: = V[t+1]; branchdual 1[tt]: = m;
     next: t := t + 1; if V[t+1] = 0
```

```
then
                     begin
                       if V[t+2] = 0
                       then go to end;
                       m := m + 1; t := t + 2
                     end:
     go to begin;
end: B := tt; M := m; K := B + 2 - M
end form branches;
procedure dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, V);
          integer K; .
          integer array branch 1, branch 2, branchdual 1, branchdual 2, V;
          integer i, j, l, h, t, search, remember;
begin
          integer array vector 1, vector 2 [1:B];
          t := 0; i := 1;
start:
         l := 1;
          for j := 1 step 1 until B do
             begin
               if branch 1[i] = i
                then
                  begin
                    vector 2[l]: = branchdual 1[j];
                    vector 1[l]: = branchdual 2[j]; l: = l + 1
                  end;
                if branch 2[i] = i
                then
```

```
- 72 -
```

```
begin
                    vector 1[l]: = branchdual 1[i];
                    vector 2[l]: = branchdual 2[j]; l: = l + 1
                  end
             end:
          t: = t + 1; V[t]: = vector 1[1]; search: = remember: = vector 2[1];
          for h := 1 step 1 until l - 1 do
begin:
             begin
               if vector 1[h] = search
                then
                  begin
                    t := t + 1; V[t] := search; search: = vector 2[h];
                    if search = remember
                    then go to continue;
                    go to begin
                  end
             end:
continue: t := t + 1; V[t] := 0; i := i + 1; if i = K + 1
                                            then go to end;
                                            go to start;
          t := t + 1; V[t] := 0
end:
end dualize;
procedure HCF(x, y); integer x, y;
begin integer RN1, RN2;
     RN1: = x; hcf: = y;
```

```
KINT := x; not
```

algorithm:

- 73 -

```
RN2: = RN1 - hcf*(RN1 \div hcf);
     if RN2 \neq 0
     then
        begin
          RN1: = hcf; hcf: = RN2;
          go to algorithm
        end:
     hcf: = abs(hcf)
end HCF;
form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M);
if K < M
then
  begin
    dualize (branch 1, branch 2, branchdual 1, branchdual 2, K, V);
    form branches (V, branch 1, branch 2, branchdual 1, branchdual 2, K, M)
  end;
comment initialize matrix INC;
begin integer i, j;
     for i := 1 step 1 until M - 1 do
         begin
           for j := i + 1 step 1 until M + i - 2 do INC [i, j] := 0
         end
end initialize matrix;
comment form upper triangle;
begin integer i:
     for i := 1 step 1 until B do
```

- 74 --

## begin

if branchdual  $2[i] \neq M$ 

then

## begin

INC [branchdual 1[*i*], branchdual 2[*i*]]: = -1;

INC [branchdual 2[i], branchdual 2[i]]: = INC [branchdual 2[i], branchdual 2[i]] + 1

75

## end;

```
INC [branchdual 1[i], branchdual 1[i]]: = INC [branchdual 1[i], branchdual 1[i]] + 1
```

#### end

```
end form upper triangle;
comment initialize inverse of INC;
begin integer i;
      for i := 1 step 1 until M - 1 do INC [i, M + i - 1] := 1
end initialize inverse of INC;
comment Gaussian elimination;
begin integer i, j, k, l, f, g, h;
      for i:=1 step 1 until M-2 do
         for j := i + 1 step 1 until M - 1 do
             begin
               if INC [i, j] \neq 0
               then
                 begin
                    HCF (INC [i,j]*INC [j,j+M-1], INC [i,i]*INC [i,i+M-1]);
                    f: = INC [i, j]*INC [j, j + M - 1] \div hcf;
                    g := INC[i, i] * INC[i, i + M - 1] \div hcf;
                    INC [j, j + M - 1]: = g*INC [j, j + M - 1];
```

```
for k := j step 1 until i + M - 1 do INC [j, k] := g * INC [j, k] - f * INC [i, k];
                    HCF (INC [i, j], INC [i, j + M - 1]);
                    for h: = i + 1 step 1 until i + M - 1 do
                       begin
                          if INC [j, h] \neq 0
                          then
                            begin
                               HCF (hcf, INC [i, h]);
                              if hcf = 1
                               then go to continue
                            end
                        end:
                    for l := j step 1 until i + M - 1 do
                        begin
                          if INC [j, l] \neq 0
                          then INC [j, l] := INC [j, l] \div hcf
                        end:
                    INC [i, j + M - 1]: = INC [i, j + M - 1] \div hcf;
                  continue:
                  end then
             end j
end Gaussian elimination;
comment calculation of the complexity;
begin integer N, D, i;
      N: = INC[1,1]; D: = INC[1,M];
      for i := 2 step 1 until M - 1 do
```

```
begin
           HCF (N*INC[i, i], D*INC[i, i + M - 1]);
           N: = INC [i, i] * N - hcf:
           D: = INC [i, i + M - 1]*D \div hcf
         end;
     complexity: = N
end calculation of the complexity;
comment backsubstitution:
begin integer i, j, k, l, m, f, g;
     for i := 1 step 1 until M - 1 do
         begin
           if INC [M-i, M-i] = complexity
           then go to for j;
           HCF(INC [M-i, M-i], complexity);
          f: = complexity \div hcf;
           g: = INC [M - i, M - i] \div hcf;
           for k := M step 1 until 2*M - i - 1 do INC [M - i, k] := INC [M - i, k]*f \div g;
        for j:
           for j:=i+1 step 1 until M-1 do
              begin
                if INC [M-j, M-i] \neq 0
                 then
                   begin
                     HCF (INC [M - i, M - i], complexity);
                     f := INC [M - j, M - i] \div hcf;
                     g:= complexity \div hcf;
```

- 77 -

```
for l := M step 1 until 2*M-1-j do INC[M-j, l] := g*INC[M-j, l]-f*INC[M-i, l];
                      if g \neq 1
                      then for m := M - j step 1 until M - 1 - i do INC [M - j, m] := g * INC [M - j, m];
                   end then
               end j
         end i
end backsubstitution:
comment put final touch to the inverse of INC;
begin integer i, j;
     for i := 1 step 1 until M - 1 do
         begin
           for j := 1 step 1 until i do ZINV [i, j] := INC [i, M-1+j]
         end lower triangle ZINV;
                                                                                                                 78
     for i := 1 step 1 until M - 1 do
         for j := i + 1 step 1 until M - 1 do ZINV [i, j] := ZINV [j, i];
     for i := 1 step 1 until M do
         begin
           ZINV [i, M]: = 0; ZINV [M, i]: = 0
         end
end final touch of the inverse of INC;
begin integer r, zero currents, RF, vertical, horizontal, b;
     Boolean original, second time, imperfection, trivial imperfection;
     integer array current, positive original, positive dual, negative original, negative dual [1:B], from original,
                   from dual [1:2*B], Bouwkamp code [1:2*B-1], address original [0:K+1], address dual
                   [0:M+1];
```

```
procedure left cyclic ordening adjacent vertices (branch 1, branch 2, branchdual 1, branchdual 2, positive,
          negative, address, from, K);
          integer K;
          integer array branch 1, branch 2, branchdual 1, branchdual 2, positive, negative, address,
                       from;
begin integer h, i, j, k, remember, meshsearch;
     k:=1; i:=1; address [0]:=0; address [1]:=1;
search first branch:
     for j := 1 step 1 until B do
         begin
           if branch 1[i] = i
           then
             begin
               remember: = meshsearch: = branchdual 1[i]; from [k]: = i; positive [i]: = k;
               go to go on searching
             end:
           if branch 2[i] = i
           then
             begin
               remember: = meshsearch: = branchdual 2[i]; from [k]: = i; negative [i]: = k;
               go to go on searching
             end
         end j;
go on searching:
         k := k + 1;
         for h := 1 step 1 until B do
```

٠

```
begin
  if branch 1[h] = i \land branchdual 2[h] = meshsearch
  then
    begin
      if branchdual 1[h] = remember
      then go to continue:
      from [k]: = h; positive [h]: = k; meshsearch: = branchdual 1[h];
      go to go on searching
    end:
  if branch 2[h] = i \wedge branchdual 1[h] = meshsearch
  then
    begin
      if branchdual 2[h] = remember
      then go to continue:
      from [k]: = h; negative [h]: = k; meshsearch: = branchdual 2[h];
      go to go on searching
    end
```

80 -

end h;

continue:

.

i:=i+1; address [i]:=k; if  $i \neq K+1$ 

then go to search first branch

end left cyclic ordening adjacent vertices;

procedure form code (branch 1, branch 2, positive, negative, address, from);

integer array branch 1, branch 2, positive, negative, address, from;

**begin integer** next first, former first, next second, former second, place, increment, signum, first vertex; integer array reduced ordered current [1:2\*B];

```
procedure reduce address (u, t):
          integer u, t;
begin
  place: = if u < address[t]
            then adress [t+1]-1
            else if u = address [t+1]
                then address [t]
                else u
end reduce address, where u is to be reduced and t the vertex;
begin integer i;
     for i := 1 step 1 until B do
         begin
           reduced ordered current [positive [i]]: = current [i] \div RF;
           reduced ordered current [negative [i]]: = -\text{current}[i] \div \text{RF}
         end i
end;
if second time
then go to first and third way of forming code;
reduce address (positive [r]+1, branch 1[r]);
next first: = abs (reduced ordered current [place]);
reduce address (positive [r]-1, branch 1[r]);
former first: = abs (reduced ordered current [place]);
reduce address (negative [r]+1, branch 2[r]);
```

next second: = abs (reduced ordered current [place]);

former second: = abs (reduced ordered current [place]);

reduce address (negative [r]-1, branch 2[r]);

```
- 81
```

```
then
        begin
          increment: = 1; go to first and third way of forming code
        end;
     if next second \geq former first \wedge next second \geq former second
     then
        begin
          increment: = 1; go to second and fourth way of forming code
        end;
     increment: = -1;
     if former first \geq former second
     then go to first and third way of forming code
     else go to second and fourth way of forming code;
first and third way of forming code:
     first vertex: = branch 1[r];
     if current [r] > 0
     then signum: = -1
     else signum : = 1;
     go to start Bouwkamp code;
second and fourth way of forming code:
     first vertex: = branch 2[r];
     if current [r] > 0
     then signum: = 1
     else signum: = -1;
start Bouwkamp code:
```

82

if next first  $\geq$  next second  $\wedge$  next first  $\geq$  former first  $\wedge$  next first  $\geq$  former second

```
begin integer end, min, where, i, j, k, l, p, q, ii, count, s, v, t, number of squares;
      integer array contour, vertex contour, save contour, save vertex contour, new squares,
                   vertices new squares [1:B];
     procedure fetch new squares (branch 1, branch 2, address, from);
                integer array branch 1, branch 2, address, from;
                comment ii has to be initialized, vertex t has to be given, b is a running variable;
     begin integer l;
           Boolean T. S:
           l:=0; T:= true; place: = address [t]; S:= true;
     label: place: = place + increment;
           reduce address (place, t)
           if reduced ordered current [place] = 0
           then go to label;
           if sign (reduced ordered current [place]) = sign (if T then signum else - signum)
           then
              begin
                if \neg (T \land (\neg S))
                then go to label;
              label 1:
                Bouwkamp code [b]: = abs (reduced ordered current [place]); b: = b + 1;
                new squares [ii]: = abs (reduced ordered current [place]);
                vertices new squares [ii]: = if branch 1 [from [place]] = t
                                             then branch 2 [from [place]]
                                             else branch 1 [from [place]];
                ii := ii + 1; go to label
              end;
```

83

```
if T \wedge S
                   then
                     begin
                       T:= false; go to label
                     end;
                  if \neg (T \land (\neg S))
                  then
                     begin
                       T: = true; S: = false; go to label 1
                     end;
                  number of squares: = ii - 1
      end fetch new squares;
      Bouwkamp code [1]: = -1; b: = 2; vertex contour [1]: = first vertex; end: =1; contour [1]:=0;
                                                                                                                  84
back: min: = contour [1]; where: = 1;
      for i := 2 step 1 until end do
          begin
            if contour [i] < \min
            then
              begin
                 \min := \operatorname{contour} [i]; \text{ where} := i
              end
         end;
      count: = 0;
      for j: = where +1 step 1 until end do
          begin
            if min \neq contour [j]
```

```
then go to next
           else count: = count + 1
         end;
next: ii := 1;
     for k := 0 step 1 until count do
         begin
           t: = vertex contour [where +k];
           fetch new squares (branch 1, branch 2, address, from)
         end;
     for i := 1 step 1 until number of squares-1 do
         begin
           if new squares [i] = new squares [i+1]
           then
             begin
               trivial imperfection: = true; go to follow
             end
         end;
follow:
     Bouwkamp code [b]: = -1; b: = b + 1;
     for l: = where + count + 1 step 1 until end do
         begin
           save contour [l]: = contour [l];
           save vertex contour [l]: = vertex contour [l]
         end;
     for p: = where step 1 until where + number of squares - 1 do
         begin
```

```
- 85 -
```

```
contour [p]: = new squares [p+1 - where] + min;
                 vertex contour [p]: = vertices new squares [p+1-where]
               end:
           for q: = where + count + 1 step 1 until end do
               begin
                 contour [number of squares - count +q-1]: = save contour [q];
                 vertex contour [number of squares - count +q-1]: = save vertex contour [q]
               end;
           if where > 1
           then s: = where - 1
           else s := 1
           for v := s + 1 step 1 until end + number of squares - count - 1 do
               begin
                 if \neg (contour [s] = contour [v] \land vertex contour [s] = vertex contour [v])
                 then
                      begin
                        s := s + 1; contour [s] := contour [v]; vertex contour [s] := vertex contour [v]
                      end
               end;
           end: = s;
           if end \neq 1
           then go to back
     end Bouwkamp code
end form code;
left cyclic ordening adjacent vertices (branch 1, branch 2, branchdual 1, branchdual 2, positive original, nega
                                    tive original, address original, from original, K);
```

left cyclic ordening adjacent vertices (branchdual 1, branchdual 2, branch 1, branch 2, positive dual, negative dual, address dual, from dual, M);

```
for r := 1 step 1 until B do
   begin integer s;
         for s := 1 step 1 until B do
             current [s]: = ZINV [branchdual 1[r], branchdual 1[s]]
                        -ZINV [branchdual 1[r], branchdual 2[s]]
                        — ZINV [branchdual 2[r], branchdual 1[s]]
                         + ZINV [branchdual 2[r], branchdual 2[s]];
         comment test imperfection;
         imperfection: = false; second time: = false; trivial imperfection: = false;
         begin integer i;
               for i := 1 step 1 until B - 1 do
                  for i := i + 1 step 1 until B do
                      begin
                        if abs (current [i]) = abs (current[i])
                         then
                           begin
                             imperfection: = true; go to count zero currents
                           end
                      end
         end;
   count zero currents:
          zero currents: = 0;
          begin integer k;
               for k := 1 step 1 until B do
```

```
begin
            if current [k] = 0
             then zero currents: = zero currents + 1
          end
end;
HCF (current [1], complexity);
begin integer l;
      for l := 2 step 1 until B do HCF (current [l], hcf);
      RF: = hcf
end;
if (complexity – current [r]) > current [r]
then
  begin
     vertical: = current [r] \div RF;
     current [r]: = current [r] - complexity;
     horizontal: = - \operatorname{current} [r] \div \operatorname{RF};
     original: = false; go to dual net
  end
else
  begin
     original: = true;
     vertical: = (complexity - current [r]) \div RF;
```

horizontal: = current  $[r] \div RF$ 

## end;

original net:

form code (branch 1, branch 2, positive original, negative original, address original, from original);

```
procedure left cyclic ordening adjacent vertices (branch 1, branch 2, branchdual 1, branchdual 2, positive,
          negative, address, from, K);
          integer K;
          integer array branch 1, branch 2, branchdual 1, branchdual 2, positive, negative, address,
                       from;
begin integer h, i, j, k, remember, meshsearch;
     k:=1; i:=1; address [0]:=0; address [1]:=1;
search first branch:
     for j := 1 step 1 until B do
         begin
           if branch 1[i] = i
           then
             begin
               remember: = meshsearch: = branchdual 1[i]; from [k]: = i; positive [i]: = k;
               go to go on searching
             end:
           if branch 2[i] = i
           then
             begin
               remember: = meshsearch: = branchdual 2[i]; from [k]: = i; negative [i]: = k;
               go to go on searching
             end
         end j;
go on searching:
         k := k + 1;
         for h := 1 step 1 until B do
```

٠

```
begin
            if current [k] = 0
             then zero currents: = zero currents + 1
          end
end;
HCF (current [1], complexity);
begin integer l;
      for l := 2 step 1 until B do HCF (current [l], hcf);
      RF: = hcf
end;
if (complexity – current [r]) > current [r]
then
  begin
     vertical: = current [r] \div RF;
     current [r]: = current [r] - complexity;
     horizontal: = - \operatorname{current} [r] \div \operatorname{RF};
     original: = false; go to dual net
  end
else
  begin
     original: = true;
     vertical: = (complexity - current [r]) \div RF;
```

horizontal: = current  $[r] \div RF$ 

## end;

original net:

form code (branch 1, branch 2, positive original, negative original, address original, from original);

go to dummy point;

dual net:

form code (branchdual 1, branchdual 2, positive dual, negative dual, address dual, from dual); dummy point:

```
punch (complexity);
punch (horizontal); punch (vertical);
if imperfection
then punch (1)
else punch (0);
punch (RF);
begin integer i;
     for i := 1 step 1 until b - 1 do punch (Bouwkamp code [i])
end:
punch (number of choices); punch (zero currents);
if ¬ imperfection
then go to next r
else if trivial imperfection
    then go to next r
    else if second time
        then go to next r
        else
           begin
             second time: = true; current [r]: = - current [r];
             if original
             then go to dual net
             else go to original net
          end;
```

```
next r:

if trivial imperfection

then punch (1)

else punch (0)

end r;

end

end;

if end of file \geq 0

then go to start

else

begin

stop; go to next B

end

end

end
```

end

# Acknowledgement

I am indebted to P. Medema who wrote the first programmes concerning the squaring problem for the I.B.M. 650. Also his help in debugging the PASCAL programmes and his assistance with the production runs are highly appreciated. The many discussions we had and his valuable remarks were a great help.

Furthermore I want to acknowledge Mr van der Sloot and Mr van der Giezen who spent several nights in taking care of part of the production runs on PASCAL.

To H. C. J. A. Nunnink I owe an example of a net (with code 145104695 4037963032730289720159810182364100) where the dual net is necessary to improve the maximum weight of the original net in the **procedure** indentification. Finally I want to thank N.V. Centrex and N.V. Eindhovensche Drukkerij for the care with which they have printed this thesis.

## Curriculum vitae

Born in the Hague, 10 Dec. 1927. H.B.S.b, Sint Janscollege, the Hague, 1946; Electrotechnisch ingenieur, Technological University, Delft, the Netherlands, 1950; Doctoraal examen, mathematics and physics, University of Amsterdam, 1955. Worked at the Mathematical Centre in Amsterdam in the computation department under the direction of prof. dr ir A. van Wijngaarden from 1953 to 1956. Joined Philips Research Laboratories, Eindhoven, the Netherlands, in 1956. Since 1960 at Philips Computing Centre.

> Address of the author: Philips Computing Centre, Eindhoven, The Netherlands.