AN ABSTRACT OF THE THESIS OF

Pat Jiun-Chang Lok for the Master of Science
(Name) (Degree)

in Mathematics presented on May 7, 1971
(Major) (Date)

Title: RECOGNITION OF PRINTED CHINESE CHARACTERS

Abstract approved: Redacted for privacy
Harry E. Goheen

This method of character recognition is according to the topological features of a given character. First store the image of a Chinese character into the storage of the computer. Each image of the character appears as a 20x20 binary matrix. Each small square in the matrix is designated as one if the reflected light is more than 50% of that of a blank point, otherwise it is zero.

The encoding method is as follows:

(A) Preprocessor: This process includes three operations. These are Cleaning, Thinning and Connecting.

(B) Preliminary Classification: First of all count all the " 1 " points in each column of the binary matrix from left to right. This list of digits is named as the Original Digit Code (ODC). From the ODC curve, by recording the extreme points, we get a Modified Digit Code (MDC).

(C) Fundamental Classification: Choosing the longest line in each column of a binary matrix from left to right form the Longest Line Code (LLC). Plot the LLC against column number, to get the LLC curve. From the LLC curve, pick up the maximum points as the Largest Digit Code (LDC) and also record the number of digits between the two largest digits in LLC as the Distance Code (DC). In order to search easily for the English translation of a given character, the assigning of the order number to each digit in LDC is more important than the LDC itself. We call these digits as the Digit Order of LDC (DOL).

According to the MDC, DC and DOL, the given character can be easily recognized by the computer.

Recognition of Printed Chinese Characters

by

Pat Jiun-Chang Lok

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

June 1971

APPROVED:

Redacted for privacy

Professor of Mathematics

in charge of major

Redacted for privacy

Acting Chairman of Department of Mathematics

Redacted for privacy

Dean of Graduate School

Date thesis is presented May 7, 1971

Typed by Barbara Eby for Pat Jiun-Chang Lok

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to Professor Goheen for his help and patient guidance.

I certainly wish to thank my dear wife, Wea-lin, and my parents for their never-ending encouragements.

TABLE OF CONTENTS

LIST OF FIGURES

RECOGNITION OF PRINTED CHINESE CHARACTERS

I. INTRODUCTION

Chinese Character Recognition

The rapid emergence of China as one of the leading producer of publications has fairly swamped United States translator monitoring Chinese activity. In nineteen hundred and sixty-two (1962) the level of Chinese to English translation was estimated at 3. 5 million words per year [1]. In contrast the estimated need by the intelligence community alone is 34. 4 million words per year. This requirement is expected to grow at the rate of about 25 million per year.

In 1960, machine translation of Chinese characters was first under taken at the University of Washington, the University of California and by the International Business Machine Corporation. So far, many methods have been used for recognizing printed Chinese characters, such as Casey and Nagy's method [2], Lam's method [3] and so on. Casey and Nagy's method uses two stages to recognize an unknown character. In the first stage, an unknown character is compared to all of the group masks [1], and a preferred order of search through the groups is defined by the mismatch scores. In the second stage, the unknown is compared in this order to the individual masks until a

[1] Each group, containing masks for a number of similar character, is represented by a single group mask.

sufficiently good match is found. Lam's method classifies the different strokes of all Chinese characters into eight types. Each character is assigned one Eight-Digit Code and one Stroke Order Code.

The method presented in here does not require a training set but needs instead only each size of a standard set of Chinese character as an input to create each corresponding Modified-Digit Code file. The Modified-Digit Code will be referred to as MDC from here on.

The simplicity of this method is that it does not require many calculations and is very efficient in locating the corresponding English translation.

<u>Chinese Character</u>

The structure of Chinese characters is usually within an imaginary rectangular frame. Most of Chinese characters consist of two parts: the radical and the phonetic. The radical imparts the meaning, while the phonetic carries the sound of the character. For example, 洲 , "continent" consists the radical 氵 , meaning "water" and the phonetic 州 , pronounced "Chou". For example, " 盲 ", "blind", consists of the radical 目 , meaning "eye" and the phonetic 亡 , pronounced "wong", meaning "die". Some radicals themselves are characters, for example, 水 is "water", 目 is "eye" and 人 is "person".

Scholars recognize two hundred and fourteen (214) radicals, but many of these have so-called "variant" forms which bear little resemblance to the "main" form. Twelve different kinds of radical in the main form and in the varient forms are given for contrast in Figure 1.

Main Form	Variant Form
人	亻
衣	衤
刀	刂
犬	犭
心	忄
牛	牜
水	氵
示	礻
火	灬
艸	艹
手	扌
肉	月

Figure 1. The first column is the main forms of radical and the second column the variant forms corresponding to the first column.

The same radical may appear in various location in a character. This is illustrated in Figure 2.

慌 忽 必 忒 忝 悶

Figure 2. All the above characters have the same radical "心" (or its variant form).

For some characters, changing radical location will not change the meaning of the characters (see Figure 3).

略 畧   旮 旯

Figure 3. Two pairs of characters. Each pair in the pairs has the same meaning.

But for some other characters, changing radical location will change the meaning of the characters (see Figure 4).

叻 另   召 叨

Figure 4. Two pairs of characters. Each pair in the pairs has a different meaning.

Character, radical and phonetic may be further analyzed in terms of "strokes" (see Figure 5).

Figure 5. Example of eight kinds of stroke.

According to the Kang Hsi Dictionary, there are 45, 000 Chinese characters. The Chinese typewriter contains 3, 580 characters. The size of the "characters" and the difficulties encountered in assigning identities to each character preclude the widespread use of typewriters and simple coding devices such as the Flexowriters.

II. INPUT CHARACTER

For recognition of Chinese printed characters, it is necessary to have a method of reading the characters mechanically. A number of methods have been considered and are presently used [4, 5]. They include optical method, such as the flying spot method, and the method of utilizing micropohones or photocell together with amplifiers, and magnetic methods such as writing with ink containing magnetic powder and using a magnetic head for reading.

The image of a Chinese character can be translated by the above mentioned methods into the storage of the computer. Each image of the character appears as a 20 x 20 binary matrix. Each small square in the matrix, which is a point, is designated as one if the reflected light is more than 50% of that of a blank point. Otherwise it is zero.

The input of printed character should be of a standard style but it is not necessarily in a fixed size, as the size can be adjusted by the amplifier. The position of a character in a binary matrix can be normalized by shifting the binary matrix through a shift register.

An example of Chinese print is shown in Figure 6.

七 成 選

Figure 6. Chinese character. This font style is commonly used in Taiwan, Hong Kong and the United States [2].

Figures 8 and 9 show the "binary" representation of some characters.

III. METHOD

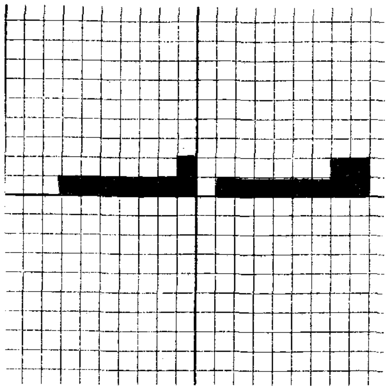
Preprocessor

In order to increase the recognition rate and reduce the reject and error rate, it is necessary to have a preprocess before the classifying of a given character.

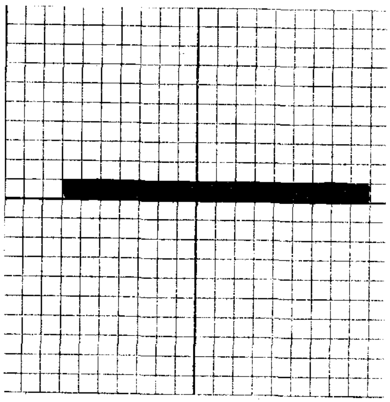
The function of the preprocessor is to make the input image be a more suitable representation. The flow of the operation is "cleaning", "thinning", "connecting". The program always does the thinning operation before the connecting operation. Otherwise the thickness of some lines can not be reduced easily. For example, there are two lines in Figure 7-A. The line in Figure 7-B is the lines in Figure 7-A after first processing the thinning operation, then the connecting operation. The line in Figure 7-C is the lines in Figure 7-A after first processing the connecting operation, then the thinning operation. A different order of operations get a different result (see Figure 7).

Cleaning. Eliminates "1" points without eight neighbors in a binary matrix. Such a point is called a stray point and would produce spikes and false connection during connect.

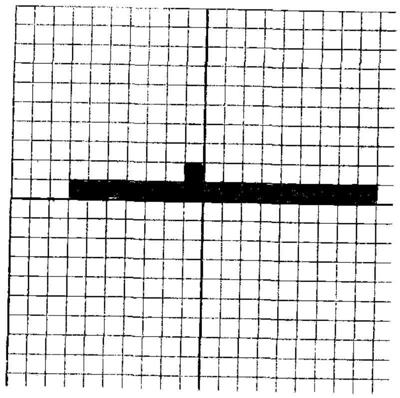
Thinning. The operation thinning reduces the thickness of the two ends of a horizontal line which consist of more than three " 1 "



A



B



C

Figure 7-A. Lines before preprocess.

7-B. Lines in Figure 7-A after first the thinning operation, then the connecting operation.

7-C. Lines in Figure 7-A after first the connecting operation, then the thinning operation.

points to make the whole line [2] more homogeneous.

Connecting. The connecting operation replaces all connecting points [3] by " 1 " points in the pattern so as to bridge the small gaps.

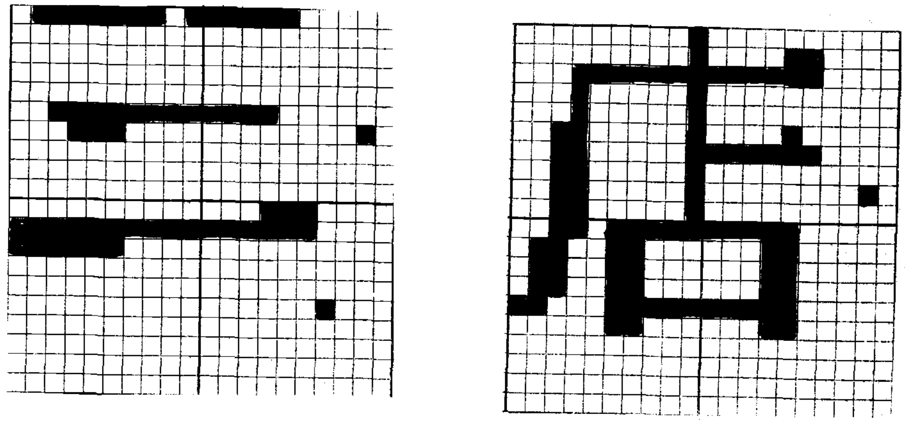


Figure 8. Badly quantized printed characters.

---

[2] For convenience, line is defined as one or more than one consecutive " 1 " points in horizontal or vertical line.

[3] The " 0 " point between horizontal lines is defined as connecting point.

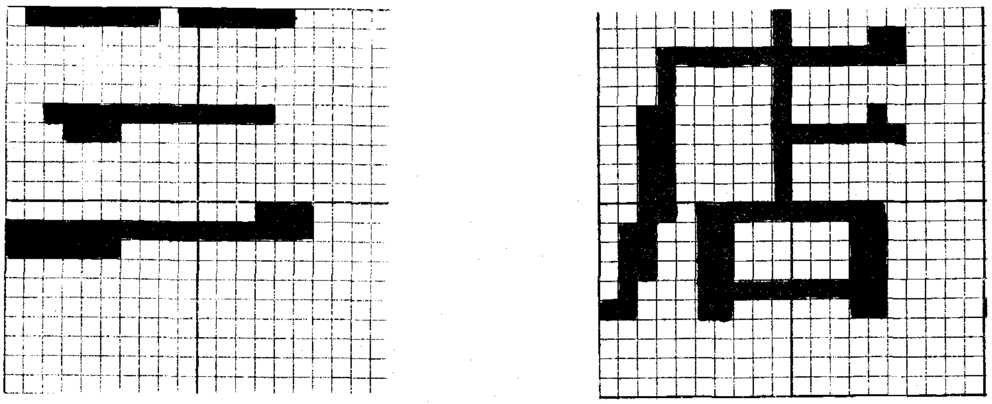


Figure 9. Characters in Figure 8 after cleaning.

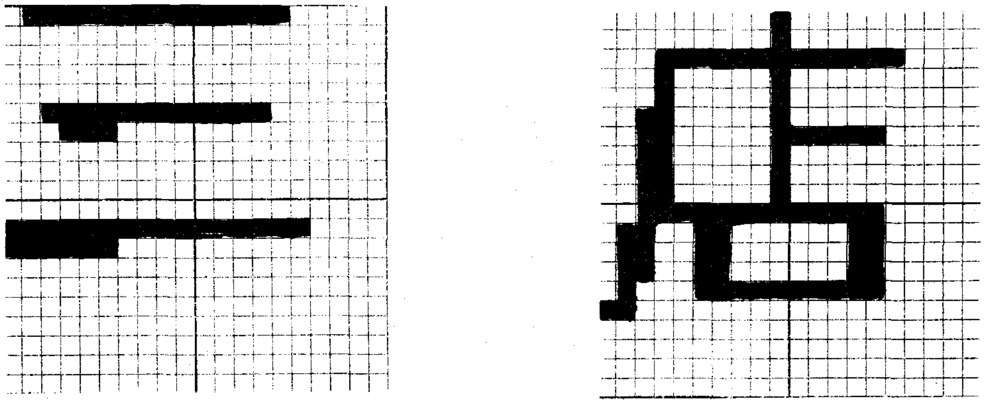


Figure 10. Characters in Figure 8 after preprocessing.

Preliminary Classification

Recognition of Chinese character is one kind of pattern recognition. The structure of Chinese character is more complicated than the characters which are used in Europe or America. The method presented here is not devised for recognizing the English alphabet.

Analysis of a character by columns in a binary matrix can be more reliable than by rows. This is because the width of a vertical stroke of Chinese character is more than three times the width of a horizontal stroke.

First of all, count all the " 1 " points in each column of the binary matrix from left to right. This list of digits is named as the Original Digit Code (ODC). ODC of 店 (right character in Figure 10) is

1 5 9 9 2 6 6 3 3 12 4 4 4 7 7 1

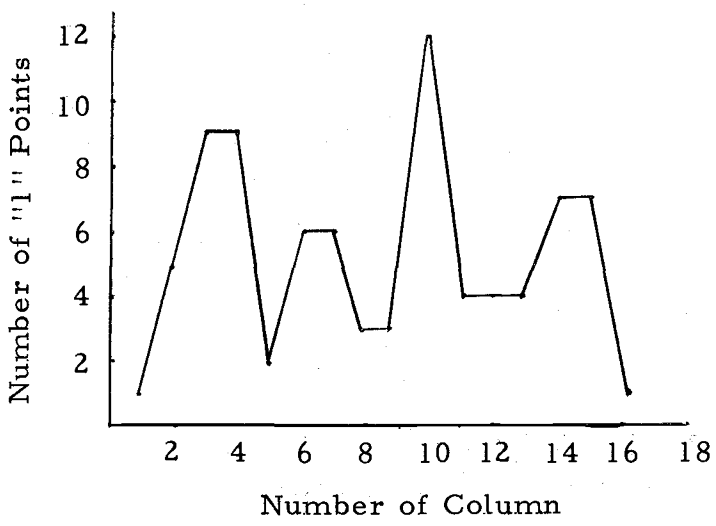


Figure 11. ODC curve of 店 .

From ODC curve (in Figure 11), by recording the extreme points, we get a Modified-Digit Code (MDC), for example, 1 9 2 6 3 12 4 7 1 as the MDC of 店 . There are nine digits in the MDC of the example. The first digit of MDC in this example is less than the second digit. So the character 店 belongs to a nine digits group with 1st digit less than 2nd digit.

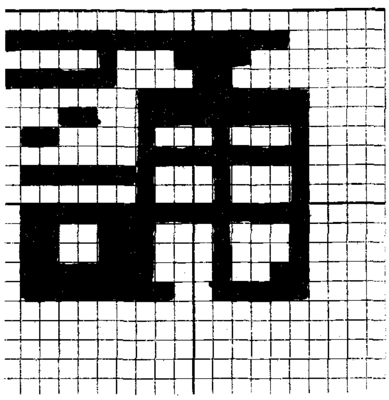


Figure 12. ODC character 誦 is
2 9 9 7 7 9 7 12 6 6 7 14 7 6 7 11,
MDC of character 誦 is
2 9 7 9 7 12 6 14 6 11 .

If ND (the number of digits in MDC) of a given character is equal 3, it is better to recalculate ODC by rows instead of by columns. Otherwise the program will have a difficult time to distinguish such characters 未 and 末 . It is because that both of them have same ND , LDC, DC and MDC . The meaning of LDC and DC will be explained in later sections. The MDC also shall be changed according to the new ODC but ND will still be 3.

Fundamental Classification

If ND is not equal three, choose the longest line in each column of a binary matrix from left to right to form the Longest-Line Code (LLC). Otherwise choose the longest line in each row of a binary matrix from top to bottom to form LLC. From LLC curve (in Figure 13), pick up the maximum points as the Largest-Digit Code (LDC) and also record the number of digits between the two largest digits in LLC as the Distance Code (DC). If two consecutive points in LLC curve are the maximum points, take the later point as a maximum point.

Examples:

LLC of 店 is 1 5 9 9 1 5 5 1 1 11 1 1 1 5 5 1

LDC of 店 is 9 5 11 5

DC of 店 is 2 2 4

LLC of 誦 is 1 5 5 2 2 5 5 11 2 2 5 14 2 2 2 11

LDC of 誦 is 5 11 14 11

DC of 誦 is 4 3 3

In order to search easily for the English translation of a given character, the assigning of the order number to each digit in LDC is more important than the LDC itself.

The following steps form a procedures to assign the order number to each digit in LDC.

1. Rearrange the digits in LDC in ascending order.
2. If the difference between two adjacent digits is less than two, the larger digit shall be replaced by the smaller one from left to right.
3. Then assign the order number to each digit. If two or three digits are the same number, they shall be assigned the same order number. The rest of the digits shall be assigned the order number according to the digit number in the rearranged list.
4. Finally, the corresponding digits in LDC are replaced by the order number and named the Digit-Order of LDC (DOL).

For example:

LDC of 店 is 9 5 11 5
DOL of 店 is 3 1 4 1

LDC of 誦 is 5 11 14 11
DOL of 誦 is 1 2 4 2

The location of a given character in the binary matrix is insubstantial as the number of digits in LDC and the digits in DC of the given character should always be the same (Figure 14).

If only the MDC and ND of the given character are computed by the computer, the computer may require a very long time to search the for corresponding English translation. This is because the MDC of the given character may be only of slight difference from the MT[4] of this character in the MDC table.

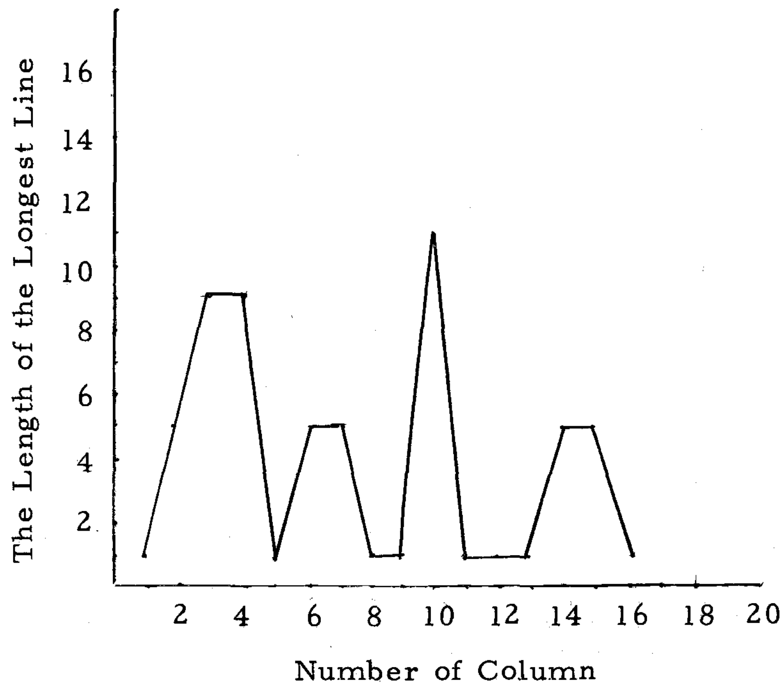


Figure 13. LLC curve of 店 .

[4] MT represents the MDC of a character in MDC table.

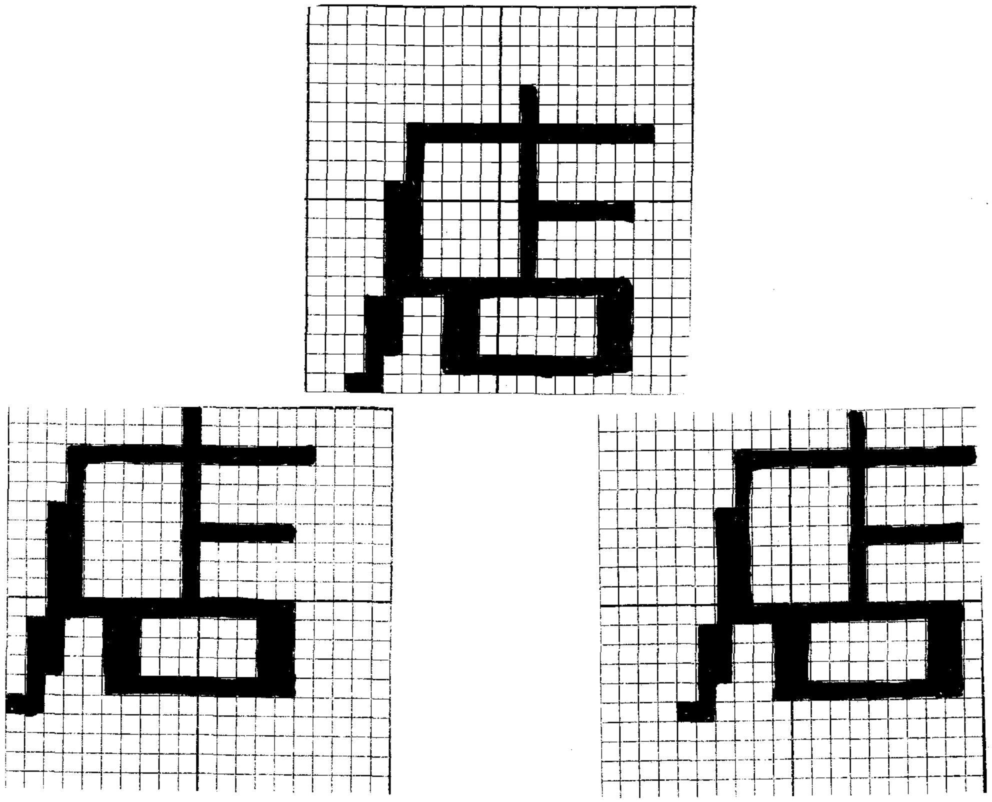


Figure 14. Same character 店 is located in the different location of a binary matrix.

A MDC File

A MDC file contains four tables (ND table, DOL table, DC table and MDC table) and subroutines which are designed to search the tables.

The way to create the tables in the MDC file is as follows:

(1) Calculate ND, MDC, DOL, and DC of each character in the set of characters which are going to be recognized.

(2) Store all the ND numbers in the ND table in ascending order of ND.

(3) Store all DOL of characters which have the same ND in the DOL table in ascending order of DOL of these characters. Then store the entry address of the DOL table and the number of these characters in the ND table.

(4) Store all DC of characters which have the same ND, DOL in the DC table in ascending order of DC of these characters. Then store the entry address of the DC table and the number of these characters in the DOL table.

(5) Store all MDC of characters which have the same ND, DOL, and DC in the MDC table and store the entry address of the MDC table in the DC table.

(6) Store the address of the corresponding translation in the MDC table.

The entrance of DOL table can be achieved by using the ND of a given character to search ND table.

Using the same technique, the entrance of DC table and MDC table can also be achieved.

Two adjacent MT in MDC table may be the same.

IV. DESCRIPTION OF THE PROGRAM

When a scanner is available, Chinese characters will be individually presented to the scanner connected to a CDC 3300 computer. However, at present, a binary matrix (20x20) is introduced manually to the computer.

The computer goes through the whole binary matrix thrice. The first time is for the preprocessor. The second time is for calculating ND, MDC. The third time is for DOL and DC.

During the preprocessor, the program is searching for lines, row by row. If a line is only a " 1 " point, and its eight neighbor points all are " 0 " points, the program is going to change this " 1 " point to be a " 0 " point.

If a line consists of more than three " 1 " points, the program is going to do a thinning process.

For convenience of describing a thinning process, we define LTP to be the line which thinned down. NLTP are lists of points which are immediately above and below the LTP line. NNL are lists of points which are immediately above the upper NLTP and immediately below the lower NLTP. The length of NLTP is the same as the length of LTP. That is, if LTP has ten points in its length, then NLTP also has ten points in its length. The length of NNL is the same as that of NLTP.

0 0 0 0 0 0 0 0 0 0 ← NNL
0 0 0 0 0 0 0 0 0 0 ← NLTP
● ● ● ● ● ● ● ● ● 0 ← LTP
0 0 0 0 0 0 0 0 0 0 ← NLTP
0 0 0 0 0 0 0 0 0 0 ← NNL

Figure 15. NNL, LTP, NLTP in a partial binary matrix.

The first three points of NLTP are called Check Points. The fourth point of NLTP is called Important Point [5]. The point directly above or below a Check Point in NNL is the Corresponding Dicision Point of that Check Point. Each Check Point has one and only one Corresponding Decision Point.

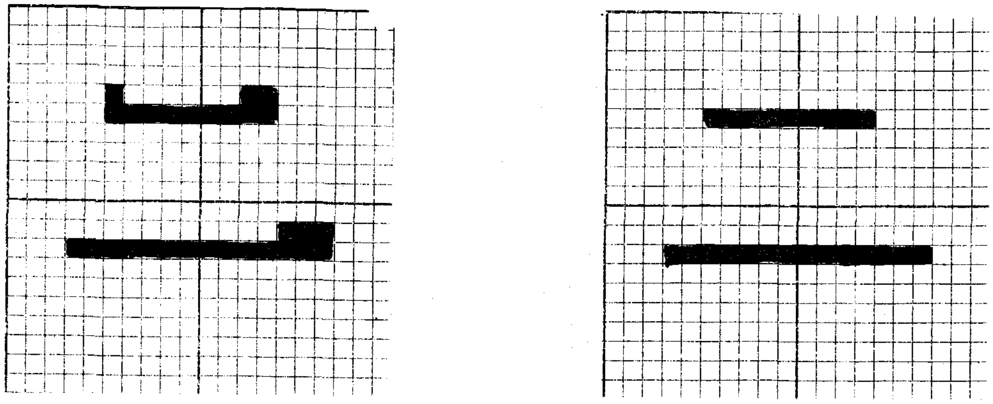
Thinning Process. There are exclusively three cases which can occur. They are the following:

1. A " 0 " Important Point precedes one or more " 1 " Check Points and the Corresponding Decision Points are all " 0 " points. In this case, all " 1 " Check Points are changed to be " 0 " points.

---

[5] Counting starts at both ends of the range of LTP, etc. For example by the fourth point of NLTP , it means this:

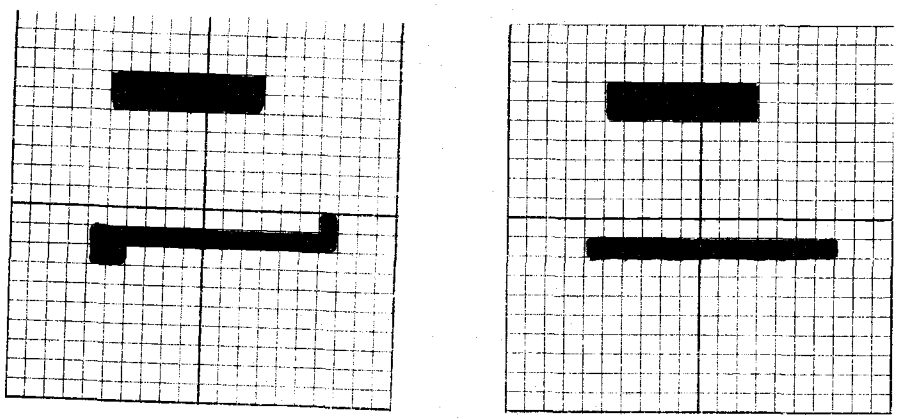
0 0 0 0 0 0 0 0 0 0 NLTP
↑ ↑
Fourth points



A B

Figure 16-A. Lines before thinning.
16-B. Lines after thinning.

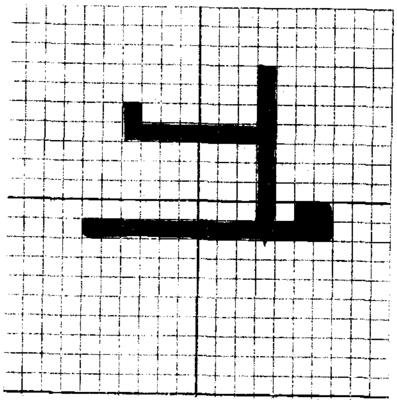
2. A " 1 " Important Points precedes all " 1 " Check Points. In this case, nothing is changed.



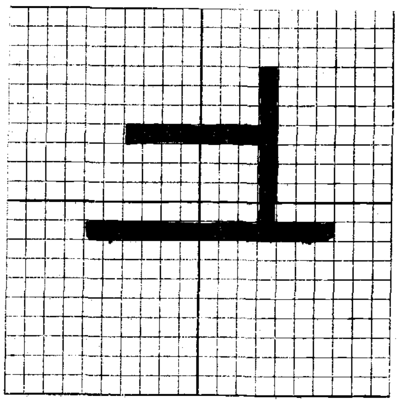
A B

Figure 17-A. Lines before thinning.
17-B. Lines after thinning.

3. The Important Point is " 1 " point followed by one or more " 0 " Check Points and finally by one or more " 1 " Check Points. In this case, the Corresponding Decision Points or Point are checked. If the Corresponding Decision Point is a " 1 " point, nothing is changed. If the Corresponding Decision Point is a " 0 " point, the Check Point is changed to be a " 0 " point.



A



B

Figure 18-A. Lines before thinning.
18-B. Lines after thinning.

All the connecting points will be changed to " 1 " points. During the classification, the program is searching for " 1 " points column by column. At the end of a search, the ODC of a given character are obtained. Then call MD subroutine, so that MDC and ND can be achieved.

In FC subroutine, first calculate the length of lines in a column (if ND is equal 3, a row shall be used instead of a column) to store these lines in TLC (i)[6], then pick up the largest digit from TLC (i) to store it in LLC (i). Repeat this process until the lengths of lines of all columns (or rows) have been calculated. From DOC subroutine, we will get DOL.

At this time, MDC, ND, DOL, and DC are printed by the computer. According to the features of a given character, such as ND, DC, DOL, the program should be able to find a very small group of MT in MDC table. The difference between the first digit of MDC (MDC (1)) of a given character and the first character of all MT(MT (1)) in this small group is less than two.

Call MT subroutine to search MDC Table, a measure of the similarity between the MDC of a given character and the MT in MDC Table is the function.

$$SSR(j) = \sum_{i=1}^{ND} (MT(i) - MDC(i))^2 \qquad j = 1 \text{ to } K$$

Where ND is the number of digits in MDC, i and j both are index. i is the digit position in MDC and the character position in MT. For example, MDC (2) is the second digit in MDC. K is the number of MT in this small group. SSR is the sum of square of

---

[6] TLC (i) are temporary storage location. i is index.

residual. The program is not going to calculate SSR (j) if the difference between MT(i) and MDC(i) is larger than two.

The MT which has the lowest value of SSR (j) is similar to the given character. If two SSR have the same lowest value, then set D equal to three, call FC and DOC subroutine to calculate LLC by rows and DOL . Finally go to search LCT table [7] to print out the corresponding English.

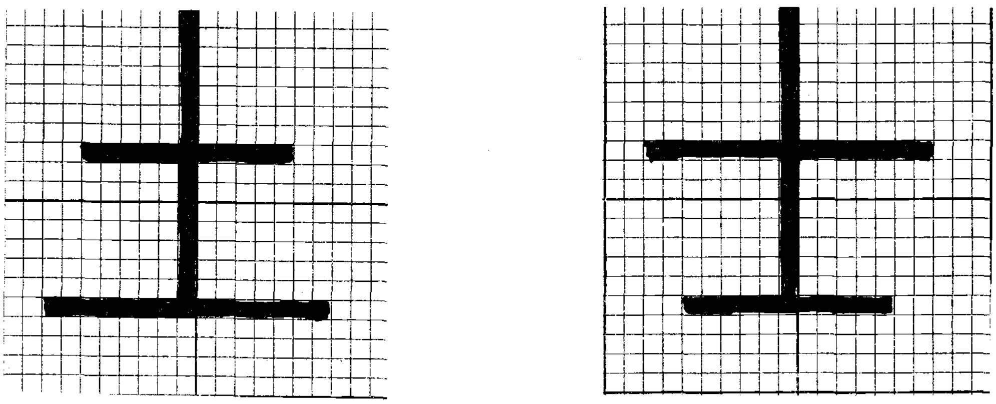
---

[7] Every item in LCT table has three words -- the first word contains DC, the second word contains DOL, the third word contains the corresponding English address.

V. DISCUSSION

In this chapter, the author would like to discuss the following: (1) why the analysis of a given character whose ND is equal to 3 is by rows instead of by columns; (2) why the sum of the squares of residuals is used to identify a given character; (3) what is the comparison of the author's method with the others mentioned in Chapter I; and (4) how good are the results of test cases.

It is very difficult to distinguish between certain characters in certain pairs by the column analysis. Examples of such complications are the pairs 土 and 士, 上 and 下, 未 and 末. Using the column analysis, two characters in the above pairs have the same ND, DOL, DC and MDC. Analyzing these characters by rows, they have different DOL, DC and MDC.



	Left	Right
DOL	1 2	2 1
DC	7	7
MDC	1 11 1 15	1 15 1 11

Figure 19. Analysis by rows of two characters which are indistinguishable by columns.

Using the sum of squares of residuals, one can make the difference between two different characters more pronounced. For example, the MDC of  is 5 3 7 3 5 and the MDC of  is 4 3 9 3 4. If the MDC of a given character is 4 3 7 3 4, it seems more reasonable to say that the given character is closer to  than to  . It is because the length of strokes (especially for the long strokes) among each other in a printed character have a roughly the same ratio. The third digit in MDC of  is one and a half times larger than the first digit. But for  , the third digit of its MDC is twice as large as the first digit. The SSR of  and the given character is 4 but the SSR of  and the given character is 2. So the program is going to decide that the given character is closer to  (with a smaller value of SSR) than to  .

By using the sum of the absolute value of the difference (SAD) between MDC and MT, the program can not decide that the given character is closer to  than to  because they have the same value of SAD. In this example, the SAD is 2.

In comparison with Mr. Lam's method, mentioned in Chapter I, the EDC method can not distinguish pairs of characters, such as 士 and 土 , 未 and 末 , 曰 and 日 . His method did not handle the problem of the preprocessor. Also it would take much computer time to determine Mr. Lam's Eight Digit Code, since the recognition of the code for each stroke may be difficult.

Casey's method [2] does not mention how to do the preprocessing. Designing good and effective group and individual masks are a tremendous and difficult job. It will take a very long time first to compare a given character with each group mask then to compare a given character with each individual mask in a certain group.

The author's method is hard to compare with Groner's method [6]. This is because his method as an aid to using a Chinese dictionary is designed for cataloging and retrieving related groups of Chinese characters having a certain given common feature.

One hundred characters have been randomly selected from the set of characters used in a Chinese typewriter. The one hundred characters each has a unique set of ND, MDC, DOL and DC codes. Also eight special pairs of characters have been selected and studied. Each member of the following five pairs can be distinguished from the other member in the pair by the row analysis of the computer program: 士 versus 土, 未 versus 末, 日 versus 曰, 千 versus 干, 刀 versus 力. Each member of the following two pairs can be distinguished from the other member in the pairs by column analysis of the computer program: 己 versus 巳, 甲 versus 由. It will take more time to distinguish 杆 from 杜, because the ND of these characters are not equal to three. Not having ND equal to three, the computer program calculates ND, DOL, DC, and MDC by column analysis. They have the same ND, DOL, DC and MDC codes. In this case, the same alarm is set as for the case ND equal to three. Using row analysis to recalculate DOL and DC of these characters, this time the program can easily distinguish 杆 from 杜 because they have different DOL and DC codes.

To save expense while the method is under consideration, the author has performed the algorithm by hand on these hundred characters. The resulting independence of location in the scanning frame of the code for each character has been gratifying.

The program will print an error message if, after a character has been preprocessed, it can not be reconciled with anything in the computer memory.

More than ten characters have been tested by this method on the machine. Average recognition time for each of these characters with the system programmed on the CDC 3300 computer was 1.8 seconds which included print out of the map of the character, the ODC, MDC, LDC, DC, DOL of the character and the corresponding translation. However, time may be reduced with modification of the program, in particularly reprogramming in machine language.

BIBLIOGRAPHY

1. Survey of the need for language translation, Planning Research Corp., IBM Survey Rept. RC-634, March 12, 1962.

2. Casey, R. and G. Nagy. Recognition of printed Chinese characters. IEEE. Vol. EC 15, No. 1, pp 91-101. February 1966.

3. Lam, P. C. M. Location and description of strokes in Chinese character by digital computer methods as part of an automatic translation of Chinese to English. O. S. U. Master Thesis, March, 1970.

4. Duff, Michael, J. B. Parallel computation in pattern recognition. Methodologies of Pattern Recognition. p. 133. 1968.

5. Tomita, Shings, Shoichi Nogvchi and Juro Oizumi. Recognition of handwritten Katakana characters. Research Institute of Electrical Communication, Tohoku University. Electronics and Communications in Japan., Vol. 50, No. 47, p. 174-182. 1967.

6. Gabriel F. Groner, J. F. Heafner and T. W. Robinson. On-Line Computer Classification of Handprinted Chinese Characters as a Translation Aid. IEEE. Vol. EC 16, No. 6, pp 856-860. December 1967.

APPENDIX

Lyapunov's Operator Scheme

II-1 H1 A2 A3 P4 A5 A6 A7 A8 H9 H10 H11 H12 P13 A14 A15 A16

II17 R18

II-1 = Read data into memory

H1 = Clean, thin, connect

A2 = Calculate ODC

A3 = Calculate MDC and ND

P4 = If ND $\neq$ 3 go to A6

A5 = Calculate MDC by rows

A6 = Set ND = 3

A7 = Calculate LDC and DC

A8 = Calculate DOL

H9 = Search ND table

H10 = Search DC table

H11 = Search DOL table

H12 = Search MDC table

P13 = If match in one MT then go to II17

A14 = Set ND = 3

A15 = Calculate DOL and DC

A16 = Search DOL and DC table

II17 = Print output

R18 = Stop

The detailed flowchart of the main program is followed by flowcharts of subroutines which have been arranged in alphabetical order. The program flow is following the direction of Arrows.

The name of a quantity or its symbol are used interchangeably throughout the flowcharts. The following abbreviations were used:

B(i, j) = An element of the binary matrix B, where $0 < i \leq 25$, $0 < j \leq 25$

C = The length of a line

CP = Index or the number of connecting points in a row

D1 = The number of digits in LDC

D2 = The distance between two adjacent points in LDC which come from LLC curve

DC = The distance code

DC(i) = The ith digit in DC

DCT = DC table

DOLT = DOL table

FL = 1, when the first digit of ODC is larger than the second digit

= 2, when the first digit of ODC is less than the second digit

II = Index or character address

IN = Index or word address

IX = Index or character address

IY = Index or word address

ID = The number of digits have been stored in MDC

MDC = Modified Digit Code

MDC(i) = The ith digit in MDC

MDCT = MDC table

MIN1 = The minimum number

MIN2 = The address of the minimum number

MT = An item in MDC table

MT(i) = The ith "character" of MT

MTL = The difference between two corresponding digits in MDC and MT

ND = The number of digits in MDC

NDT = ND table

NL = The number of digitis in LLC

ODC = Original Digit Code

ODC(i) = The ith digit in ODC

S = 0, there is one minimum number

S = 1, there are two minimum numbers

S1 = 0, while searching forward in MDC table from the entrance

S1 = 1, while searching backward in MDC table from the entrance

SS = The total of " 1 " points in a colum or a row

Tj = The temporary store for j

TLC = Temporary store

TLC(i) = The ith digit in TLC

The function of subroutines are listed as following:

DOC = Calculate DOL

FC = Calculate TLC, LLC and LDC

MD = Calculate MDC and ND

MT = Search MDC table to measure a similarity between the MDC of a given character and MT in MDC table and print the corresponding English or print error message, or return with the entrance address of LCT table

SEARCH = Search ND table, DC table and DOL table, return with the entrance address (stored in IN) of next table or error message

SRLCT = Search LCT table and print the corresponding English

Tables:

IN = Index, 15 bits

NI = Number of items in a group, 9 bits

ND Table

1 word per item

First 9 bits contain NI

Next 15 bit for IN

```
        01200007
NI                  IN
        01400017
```

DC Table

2 words per item

First word for DC

```
        00000314    DC

NI      01600021    IN
```

DOL Table

2 words per item

First word for DOL

```
        00000132    DOL

NI      00700021    IN
```

LCT Table

3 words per item

First word for DC

Second word for DOL

Third word for IN

```
00000017    DC

00001422    DOL

00001202    IN
```

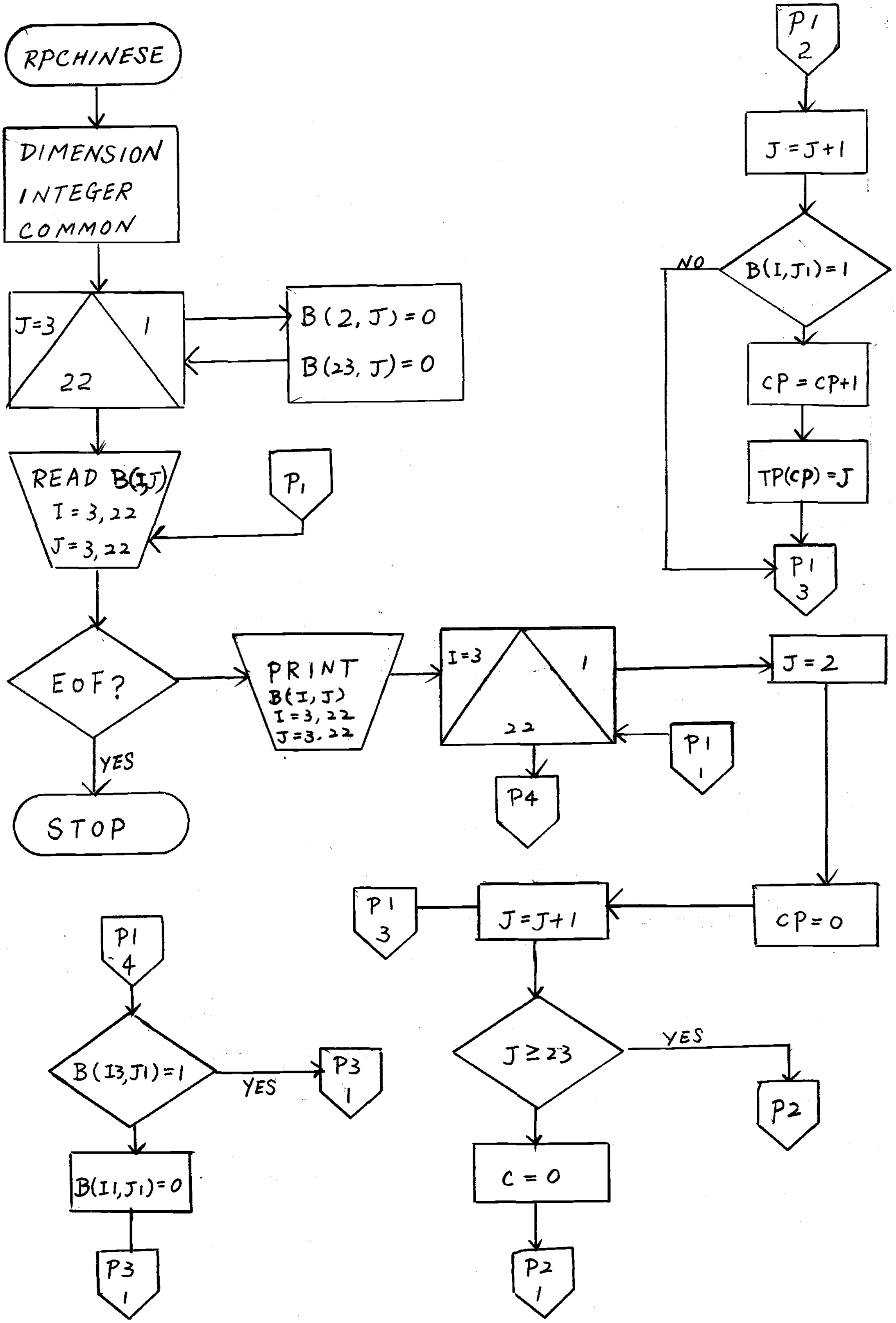
MDC Table

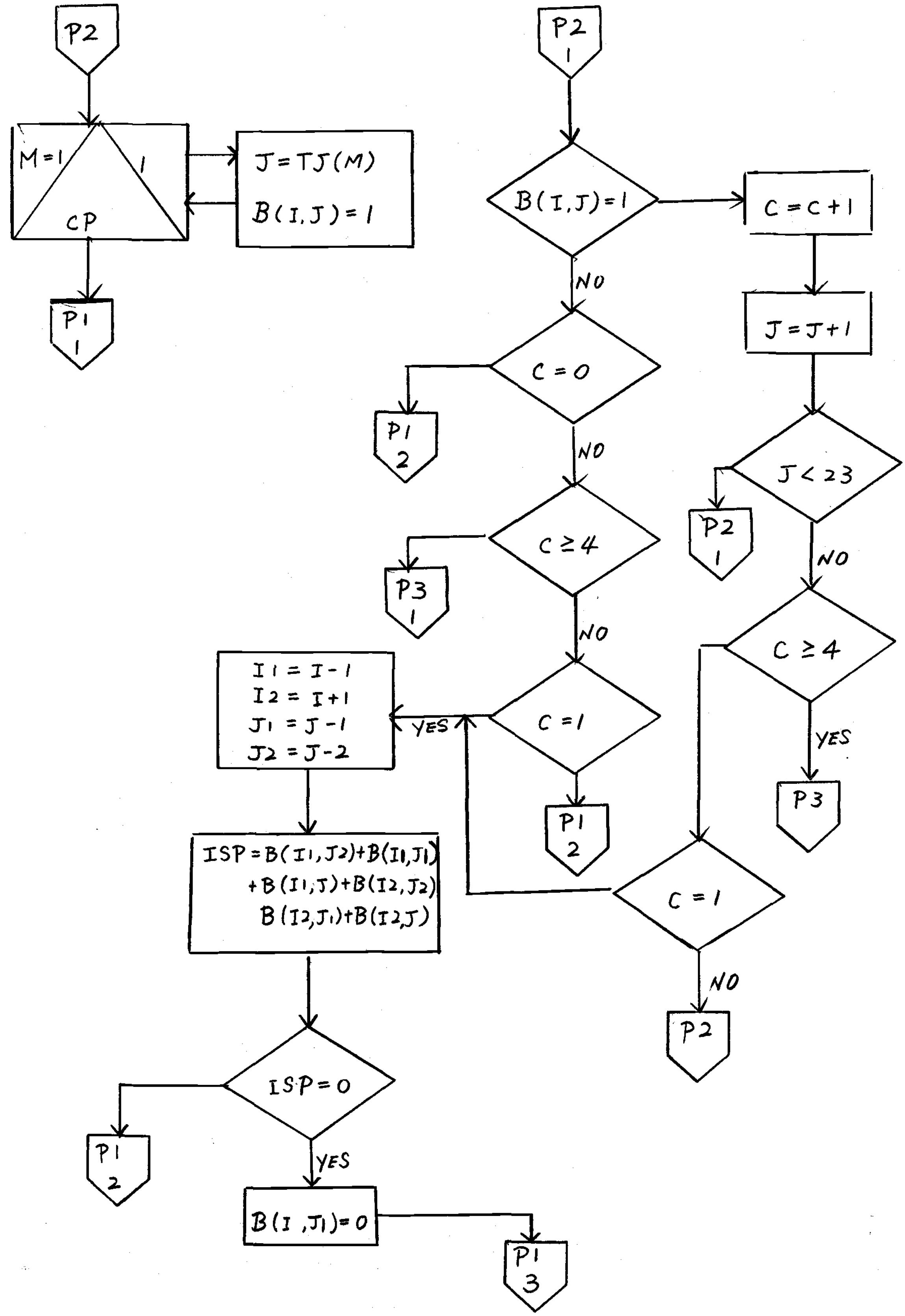
3 word per item

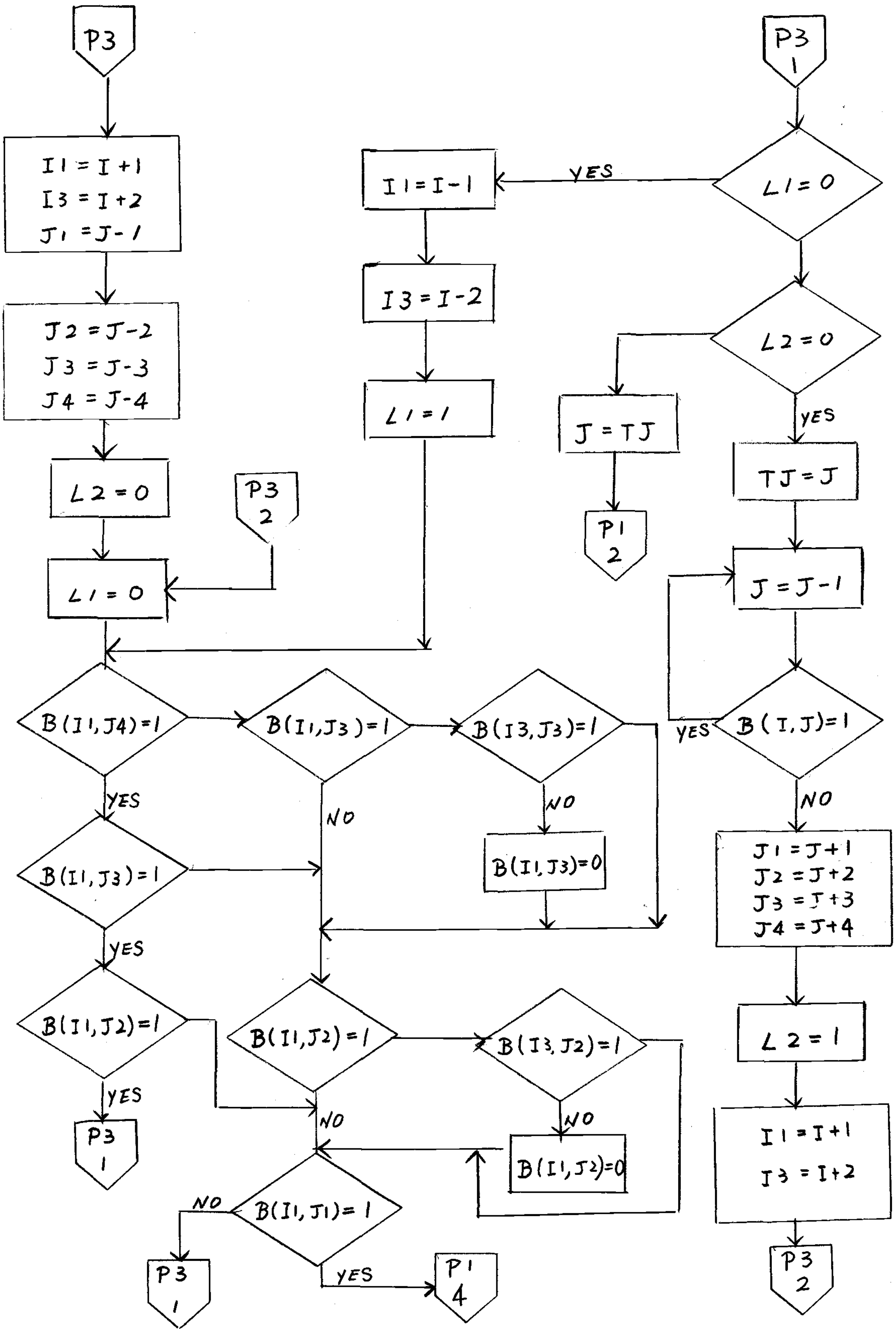
First two words for the first eight digits of MDC

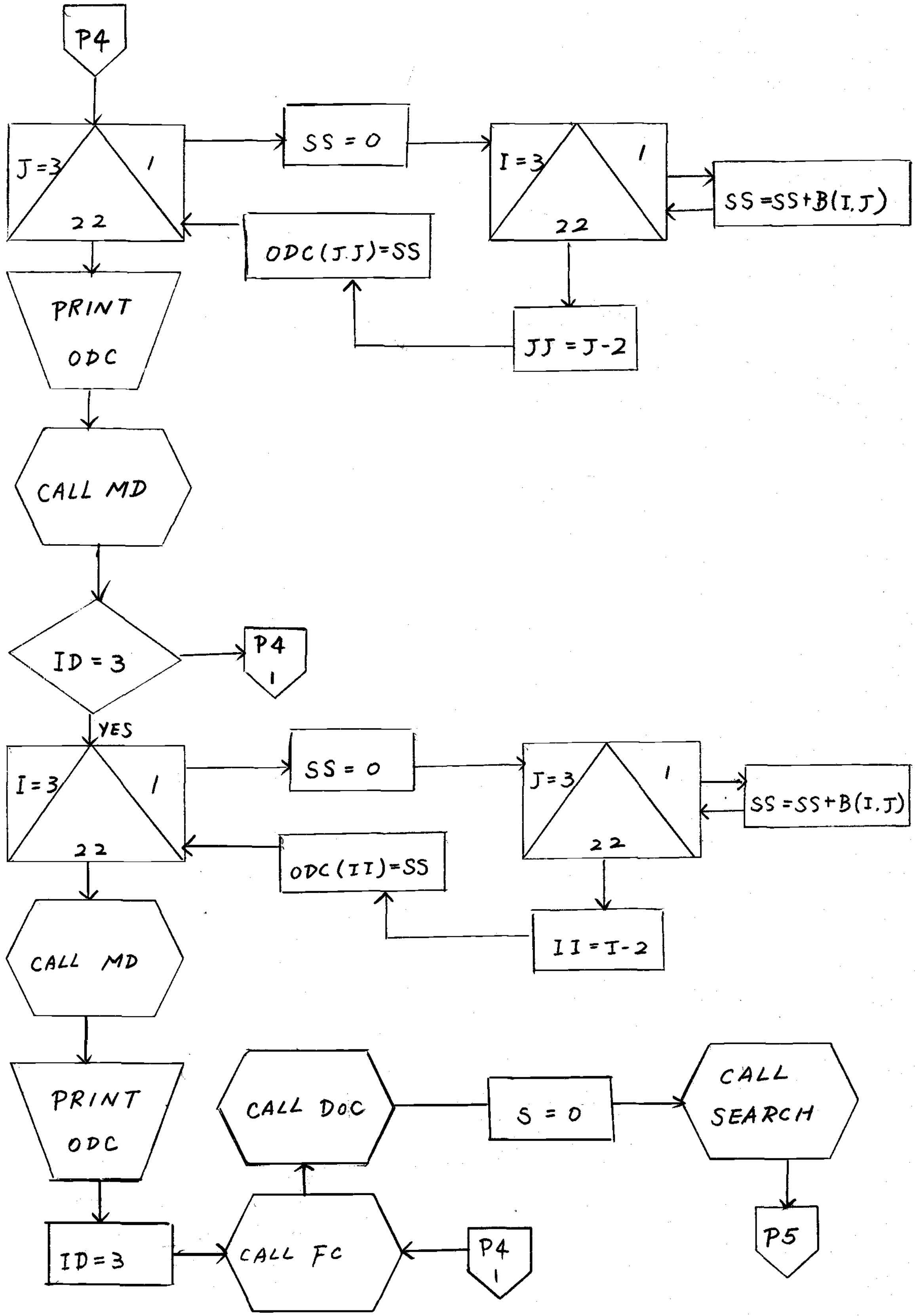
15 bits for address

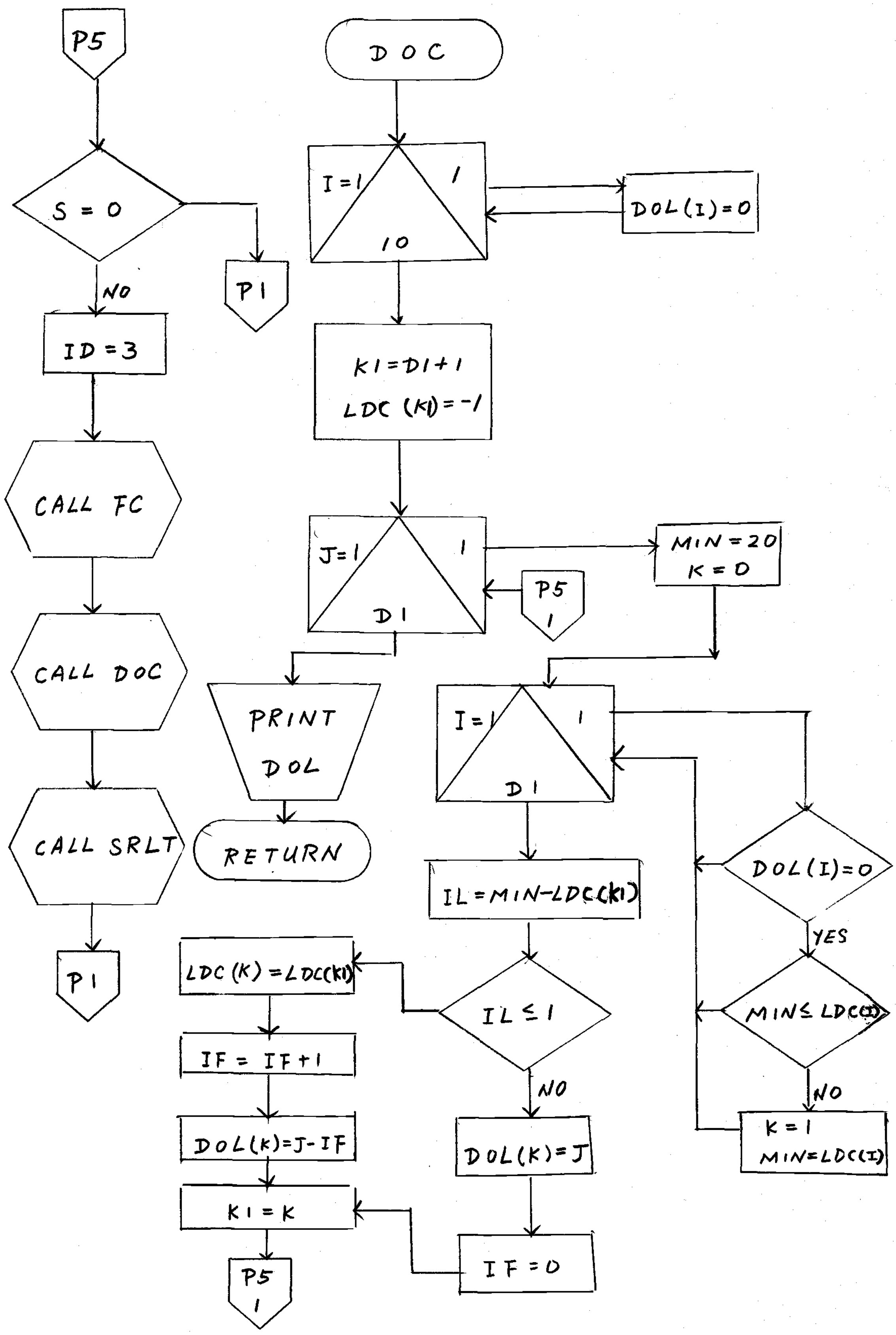
```
03060207 }
02110204 } MT
00045212    address
```

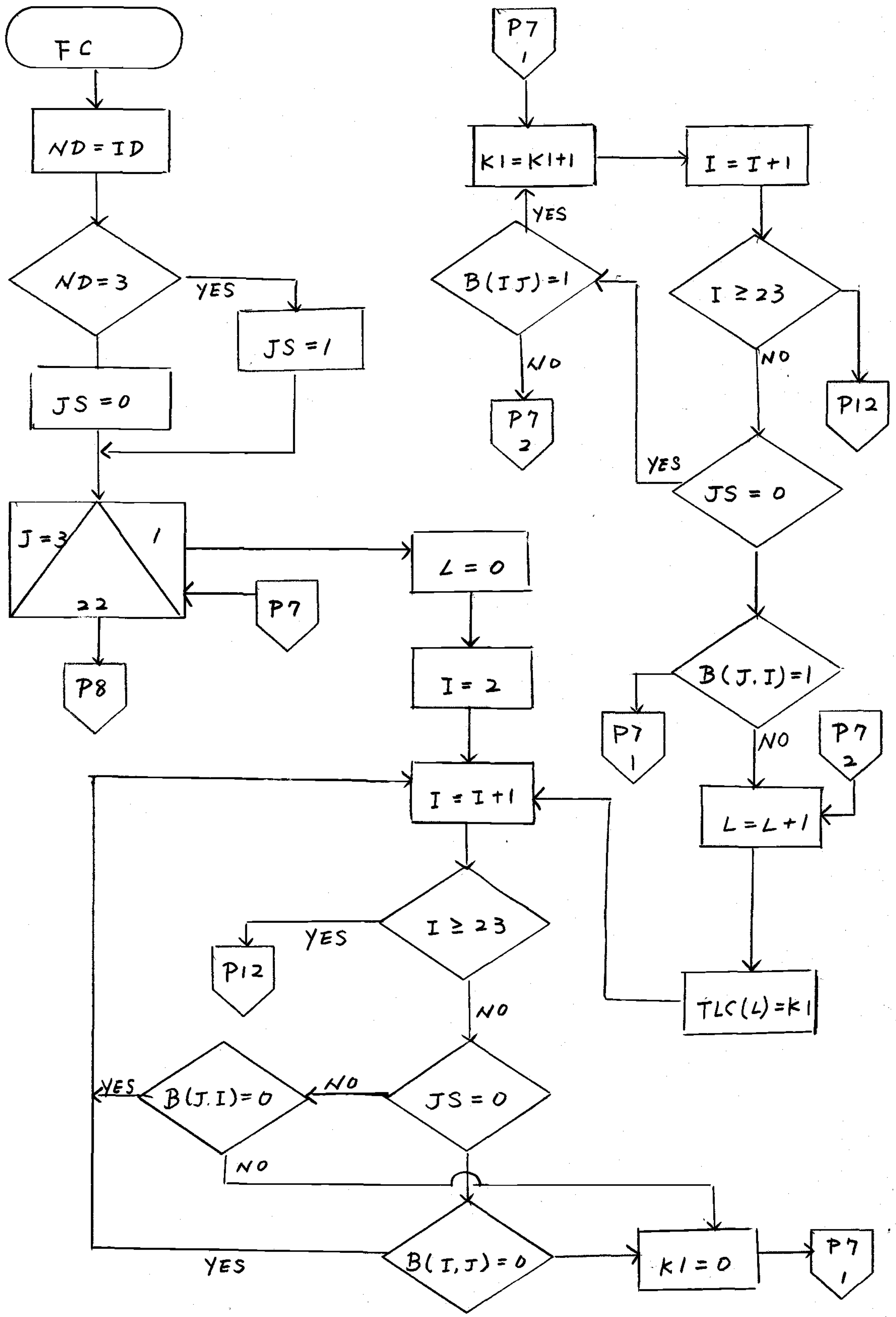

RPCHINESE
DIMENSION
INTEGER
COMMON
J=3
1
22
B(2,J)=0
B(23,J)=0
READ B(I,J)
I=3,22
J=3,22
P1
EOF?
YES
STOP
PRINT
B(I,J)
I=3,22
J=3,22
I=3
1
22
P4
P1
1
J=2
CP=0
J=J+1
P1
3
J≥23
YES
P2
C=0
P2
1
P1
2
J=J+1
B(I,J1)=1
NO
CP=CP+1
TP(CP)=J
P1
3
P1
4
B(I3,J1)=1
YES
P3
1
B(I1,J1)=0
P3
1


P2
M=1
1
CP
J=TJ(M)
B(I,J)=1
P1 1
P2 1
B(I,J)=1
C=C+1
NO
J=J+1
C=0
P1 2
NO
J<23
P2 1
C≥4
P3 1
NO
NO
C≥4
I1=I-1
I2=I+1
J1=J-1
J2=J-2
YES
C=1
YES
P3
P1 2
ISP=B(I1,J2)+B(I1,J1)
+B(I1,J)+B(I2,J2)
B(I2,J1)+B(I2,J)
C=1
NO
P2
ISP=0
P1 2
YES
B(I,J1)=0
P1 3

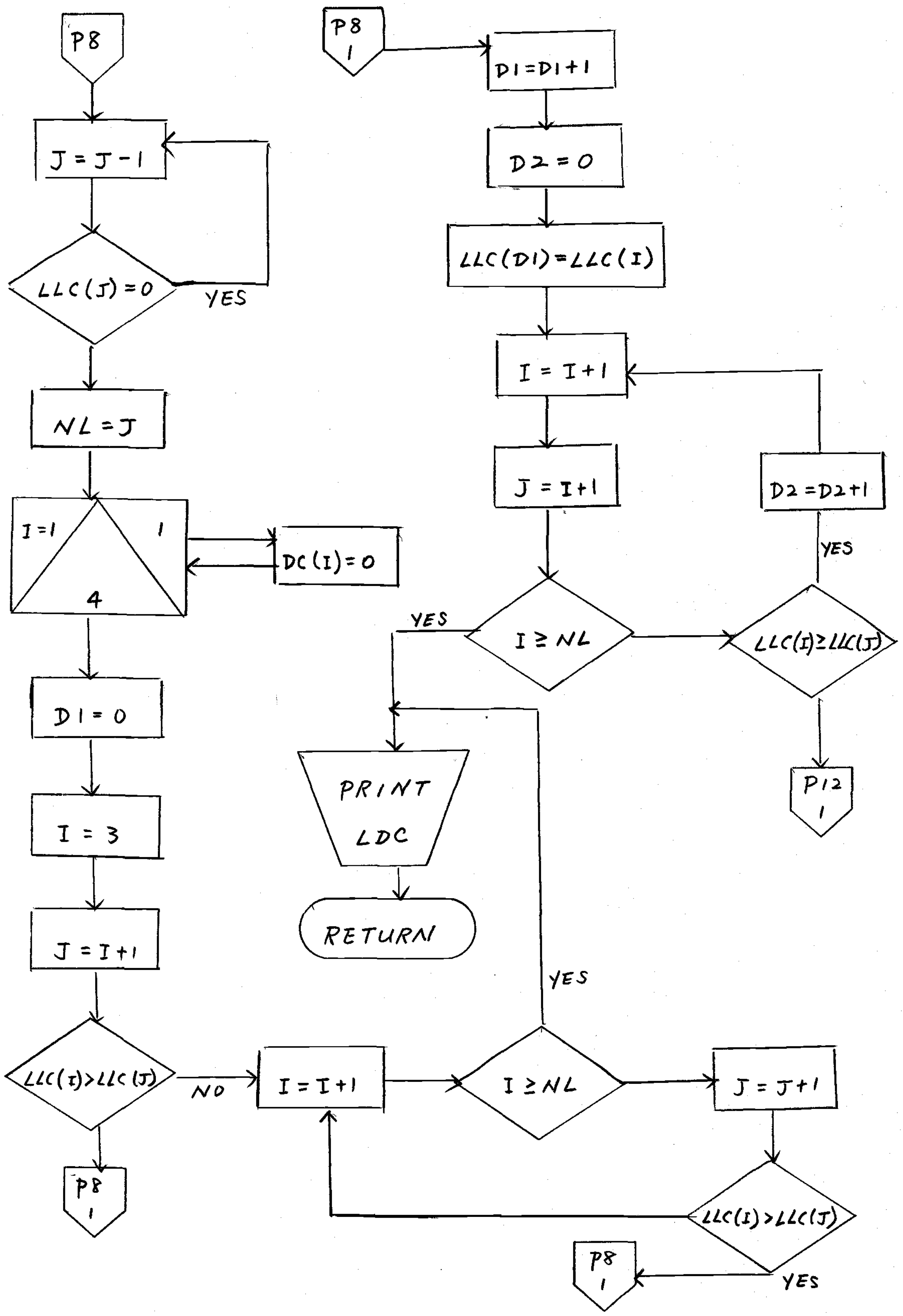

P3
I1 = I + 1
I3 = I + 2
J1 = J - 1
J2 = J - 2
J3 = J - 3
J4 = J - 4
L2 = 0
P3 2
L1 = 0
I1 = I - 1
YES
I3 = I - 2
L1 = 1
P3 1
L1 = 0
L2 = 0
J = TJ
P1 2
YES
TJ = J
J = J - 1
YES
B(I,J) = 1
NO
J1 = J + 1
J2 = J + 2
J3 = J + 3
J4 = J + 4
L2 = 1
I1 = I + 1
I3 = I + 2
P3 2
B(I1,J4) = 1
YES
B(I1,J3) = 1
YES
B(I1,J2) = 1
YES
P3 1
B(I1,J3) = 1
NO
B(I3,J3) = 1
NO
B(I1,J3) = 0
B(I1,J2) = 1
B(I3,J2) = 1
NO
B(I1,J2) = 0
NO
B(I1,J1) = 1
NO
P3 1
YES
P1 4

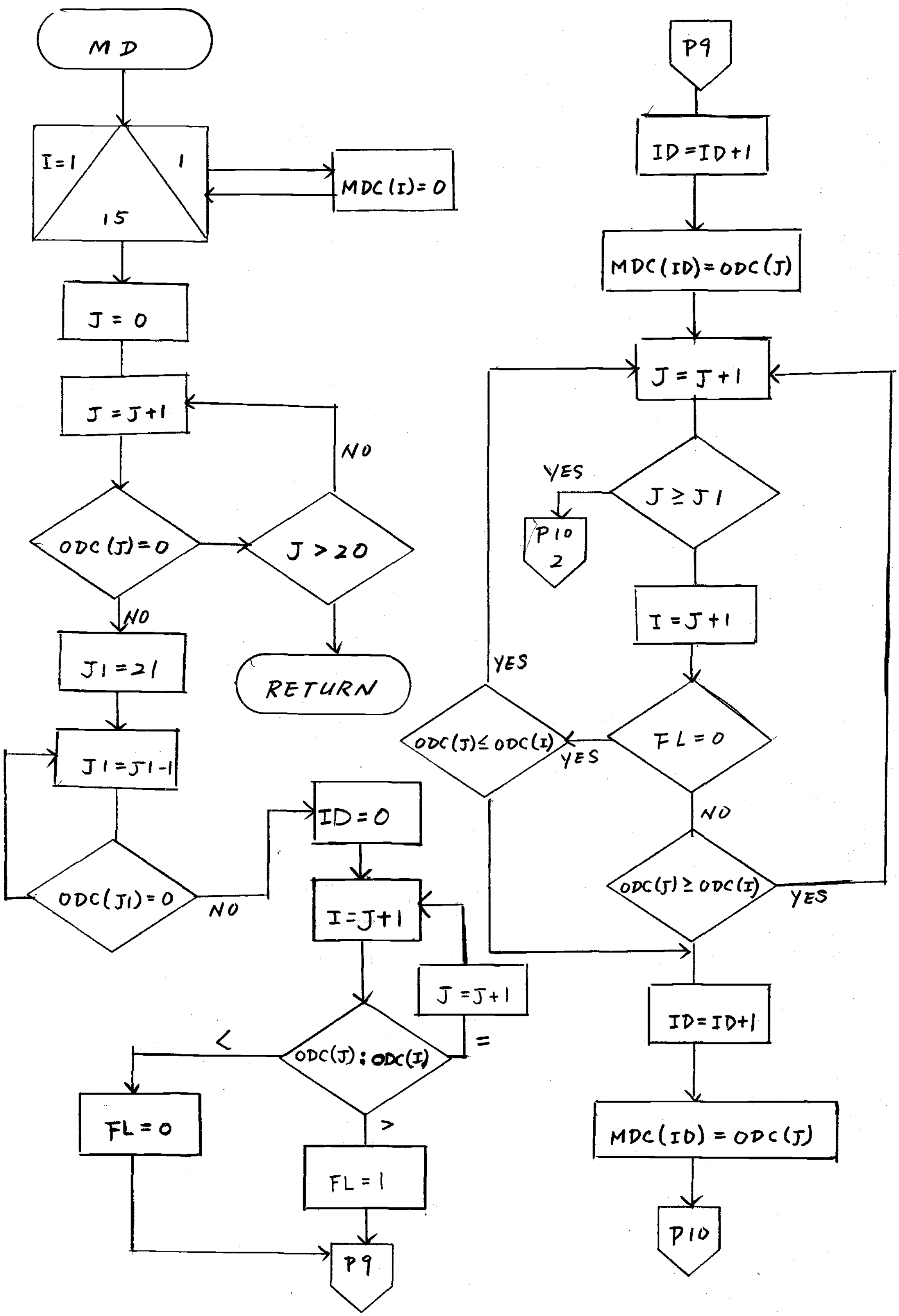

P4
J=3
1
22
SS = 0
I=3
1
22
SS =SS+B(I,J)
ODC(JJ)=SS
JJ = J-2
PRINT
ODC
CALL MD
ID = 3
P4
1
YES
I=3
1
22
SS = 0
J=3
1
22
SS =SS+B(I,J)
ODC(II)=SS
II = I-2
CALL MD
PRINT
ODC
CALL DOC
S = 0
CALL
SEARCH
P5
ID=3
CALL FC
P4
1

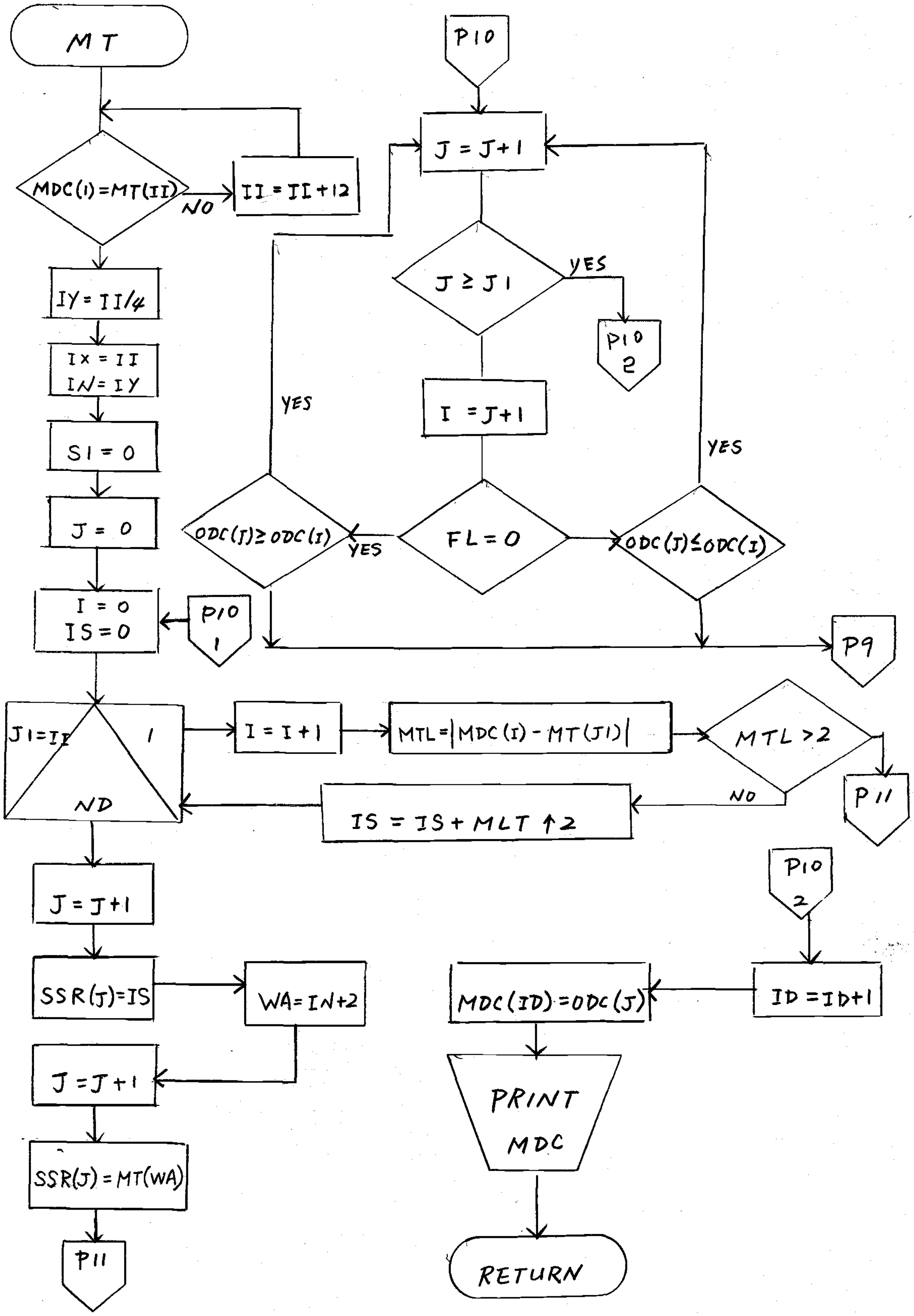

P5
S = 0
NO
P1
ID = 3
CALL FC
CALL DOC
CALL SRLT
P1
DOC
I = 1
1
10
DOL(I) = 0
K1 = D1 + 1
LDC (K1) = -1
J = 1
1
D1
MIN = 20
K = 0
P5
1
PRINT
DOL
RETURN
I = 1
1
D1
IL = MIN - LDC(K1)
DOL(I) = 0
YES
MIN ≤ LDC(I)
NO
K = I
MIN = LDC(I)
LDC (K) = LDC(K1)
IL ≤ 1
NO
IF = IF + 1
DOL(K) = J - IF
DOL(K) = J
K1 = K
IF = 0
P5
1

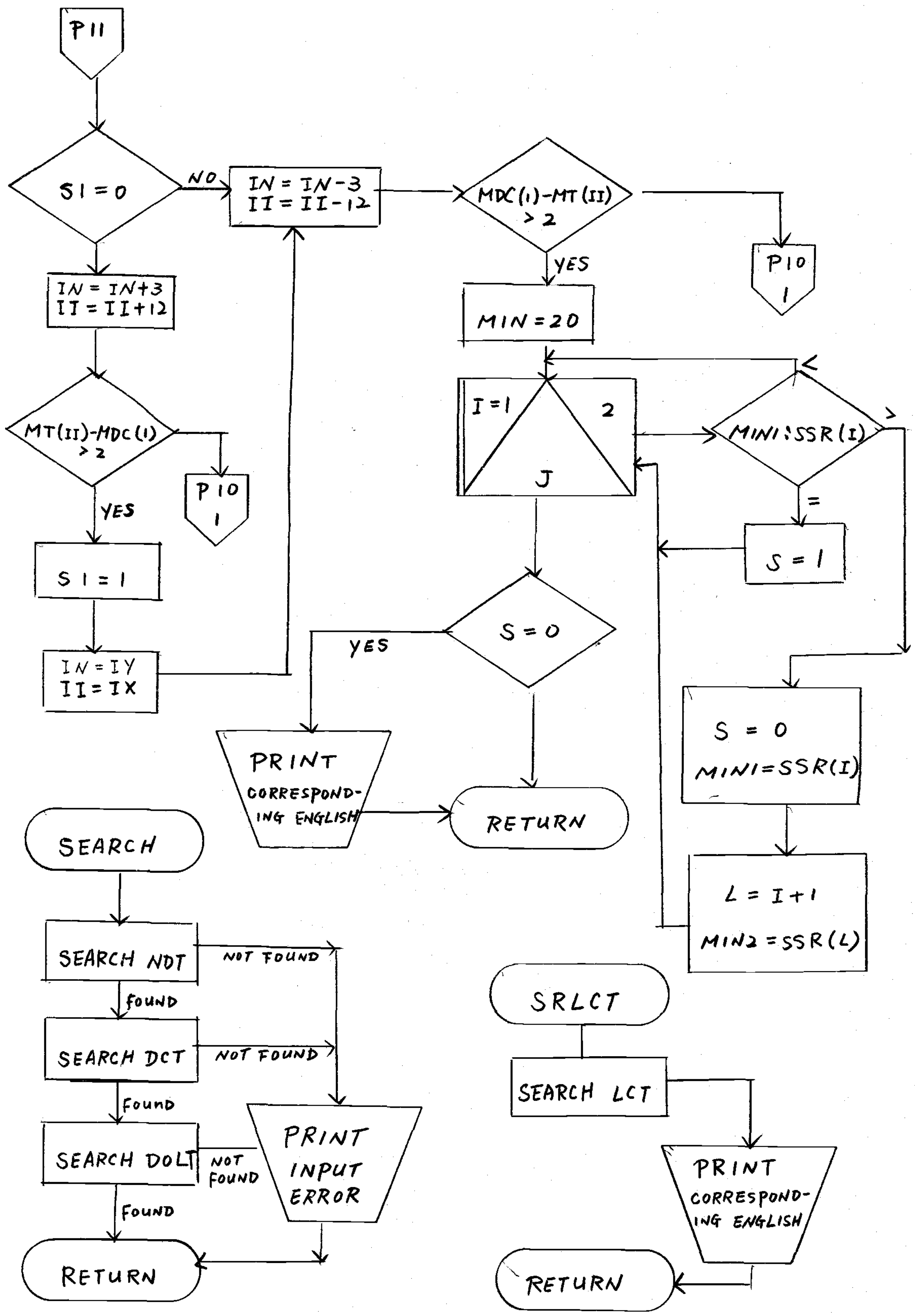


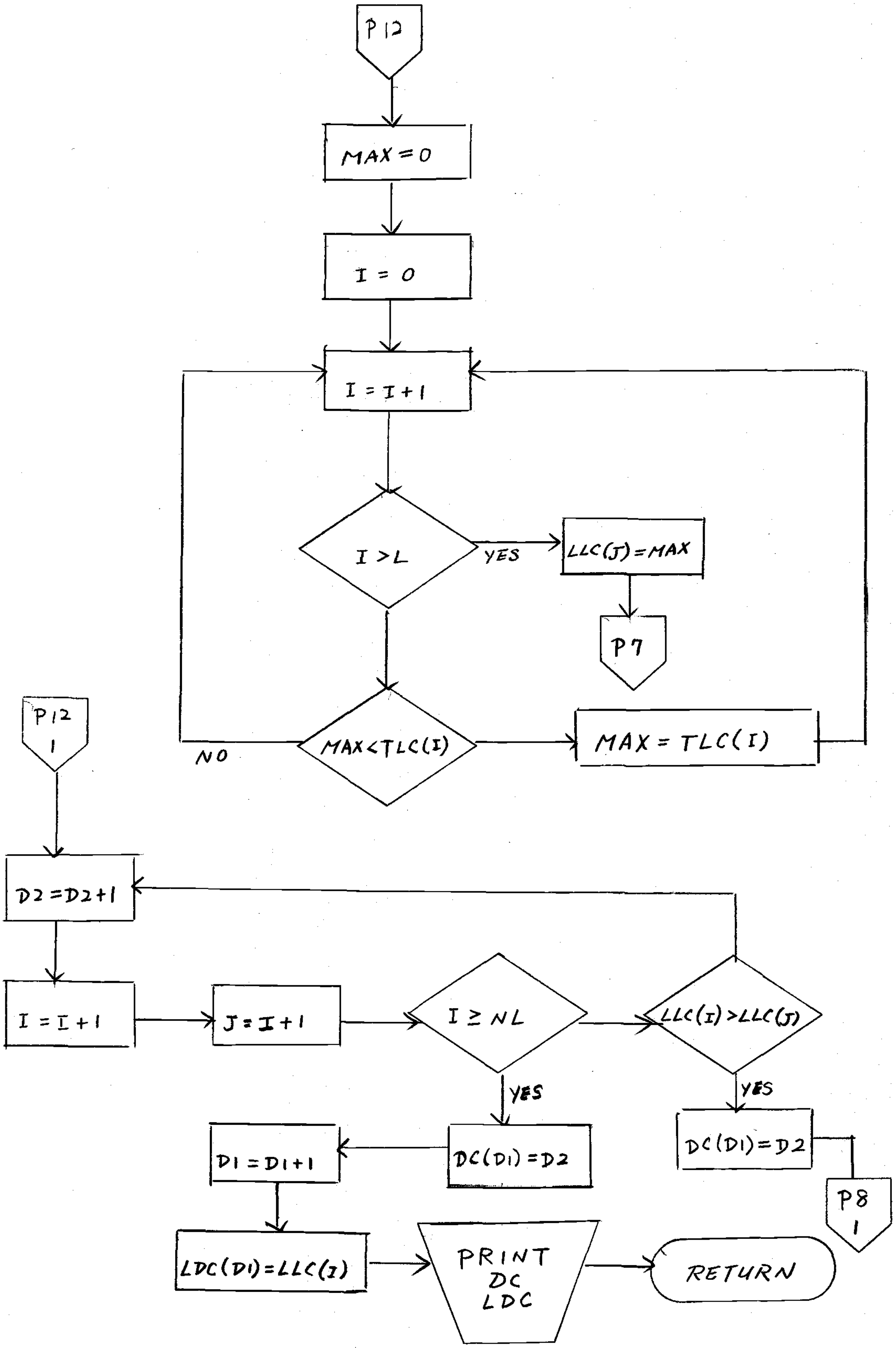
FC
ND = ID
ND = 3
YES
JS = 1
JS = 0
J = 3
1
22
P7
P8
L = 0
I = 2
I = I + 1
I ≥ 23
YES
P12
NO
JS = 0
NO
B(J,I) = 0
YES
NO
B(I,J) = 0
YES
K1 = 0
P7
1
P7
1
K1 = K1+1
I = I+1
YES
B(I J) = 1
NO
P7
2
I ≥ 23
NO
P12
YES
JS = 0
B(J,I) = 1
P7
1
NO
P7
2
L = L + 1
TLC(L) = K1


P8
J = J - 1
LLC(J) = 0
YES
NL = J
I=1
1
4
DC(I) = 0
D1 = 0
I = 3
J = I+1
LLC(I) > LLC(J)
NO
P8
1
P8
1
D1=D1+1
D2 = 0
LLC(D1) = LLC(I)
I = I+1
J = I+1
D2 = D2+1
YES
YES
I ≥ NL
LLC(I) ≥ LLC(J)
P12
1
PRINT
LDC
RETURN
YES
I = I+1
I ≥ NL
J = J+1
LLC(I) > LLC(J)
P8
1
YES


MD
I=1
1
15
MDC(I)=0
J = 0
J = J+1
NO
ODC(J)=0
J > 20
NO
J1 = 21
RETURN
J1 = J1-1
ODC(J1) = 0
NO
ID = 0
I = J+1
J = J+1
=
<
ODC(J) : ODC(I)
>
FL = 0
FL = 1
P9
P9
ID = ID+1
MDC(ID) = ODC(J)
J = J+1
YES
J ≥ J1
P10
2
I = J+1
YES
ODC(J) ≤ ODC(I)
YES
FL = 0
NO
ODC(J) ≥ ODC(I)
YES
ID = ID+1
MDC(ID) = ODC(J)
P10


MT
MDC(I)=MT(II)
NO
II = II+12
IY = II/4
IX = II
IN = IY
SI = 0
J = 0
I = 0
IS = 0
P10
1
J1=II
1
ND
I = I+1
MTL = |MDC(I) - MT(J1)|
MTL > 2
NO
P11
IS = IS + MLT ↑ 2
J = J+1
SSR(J)=IS
WA = IN+2
J = J+1
SSR(J) = MT(WA)
P11
P10
J = J+1
J ≥ J1
YES
P10
2
I = J+1
FL = 0
YES
ODC(J) ≥ ODC(I)
YES
ODC(J) ≤ ODC(I)
YES
P9
P10
2
ID = ID+1
MDC(ID) = ODC(J)
PRINT
MDC
RETURN


P11
S1 = 0
NO
IN = IN-3
II = II-12
MDC(I)-MT(II) > 2
P10 1
YES
IN = IN+3
II = II+12
MIN = 20
I = 1
2
J
MIN1 : SSR(I)
<
>
=
MT(II)-MDC(I) > 2
P10 1
YES
S1 = 1
S = 1
S = 0
YES
IN = IY
II = IX
S = 0
MIN1 = SSR(I)
PRINT CORRESPONDING ENGLISH
RETURN
SEARCH
L = I+1
MIN2 = SSR(L)
SEARCH NDT
NOT FOUND
FOUND
SRLCT
SEARCH DCT
NOT FOUND
SEARCH LCT
FOUND
PRINT INPUT ERROR
SEARCH DOLT
NOT FOUND
PRINT CORRESPONDING ENGLISH
FOUND
RETURN
RETURN


P12
MAX = 0
I = 0
I = I + 1
I > L
YES
LLC(J) = MAX
P7
MAX < TLC(I)
NO
MAX = TLC(I)
P12
1
D2 = D2 + 1
I = I + 1
J = I + 1
I ≥ NL
YES
LLC(I) > LLC(J)
YES
DC(D1) = D2
P8
1
D1 = D1 + 1
DC(D1) = D2
LDC(D1) = LLC(I)
PRINT
DC
LDC
RETURN

```
      PROGRAM RPCHINESE
      DIMENSION OC(10),OOL(10),MOC(15),OOC(20)
      DIMENSION TJ(20),LOC(20),B(25,25)
      INTEGER B,OC,OOL,LOC,MOC,OOC,O1,IO
      INTEGER TJ,CP,C,SS,S,O1
      COMMON OC,OOL,MOC,O1,IO,MIN2,S,OOC,LOC,B
C     ***INPUT DATA ***
      DO 996 J=3,22
      B(2,J)=0
  996 B(23,J)=0
  999 READ  (6,100) ((B(I,J),J=3,22),I=3,22)
  100 FORMAT (20I1)
      IF (EOF(6)) GO TO 101
      WRITE (61,98) ((B(I,J),J=3,22),I=3,22)
   98 FORMAT (1H ,20(I2,2X))
      DO 1 I=3,22
      J=2
      CP=0
    2 J=J+1
      IF ( J .GE. 23 ) GO TO 10
      C=0
   11 IF (B (I,J) .EQ. 1 ) GO TO 5
      IF ( C .EQ. 0 ) GO TO 2
      IF (C .GE. 4 ) GO TO 99
      IF  ( C .NE. 1 ) GO TO 3
C     *** IF THERE IS A NOISE , GET RID OF IT***
   25 I1=I-1
      I2=I+1
      J1=J-1
      J2=J-2
      ISP=B(I1,J2)+B(I1,J1)+B(I1,J)+B(I2,J2)
      ISP=ISP+B(I2,J1)+B(I2,J)
      IF ( ISP .NE. 0 ) GO TO 3
      B(I,J1)=0
      GO TO 2
C     *** CHECK CONNECTING POINT***
    3 J1=J+1
      IF ( B(I,J1) .NE. 1 ) GO TO 2
      CP=CP+1
      TJ(CP)=J
      GO TO 2
    5 C=C+1
      J=J+1
      IF ( J .LT. 23) GO TO 11
      IF ( C .GE. 4) GO TO 99
      IF ( C .EQ. 1 ) GO TO 25
C      **CONNECTING PROCESS ***
   10 DO 12 M=1,CP
      J=TJ(M)
   12 B(I,J)=1
      GO TO 1
C     *** THINNING PROCESS ***
   99 I1=I+1
      I3=I+2
      J1=J-1
      J2=J-2
      J3=J-3
      J4=J-4
      L2=0
   16 L1=0
   15 IF ( B(I1,J4) .EQ. 1 ) GO TO 17
      IF ( B(I1,J3) . NE. 1 ) GO TO 18
      IF ( B(I3,J3) .EQ.  1 ) GO TO 18
      B(I1,J3)=0
      GO TO 18
   17 IF ( B(I1,J3) .EQ. 1 ) GO TO 19
   18 IF  ( B(I1,J2) .NE . 1) GO TO 20
      IF ( B(I3,J2) .EQ. 1) GO TO 20
      B(I1,J2)=0
      GO TO 20
   19 IF ( B(I1,J2) .EQ. 1 ) GO TO 13
   20 IF ( B(I1,J1) .NE. 1 ) GO TO 13
      IF ( B(I3,J1) .EQ. 1 ) GO TO 13
      B(I1,J1)=0
   13 IF ( L1 .NE. 0) GO TO 22
      I1=I-1
      I3=I-2
      L1=1
      GO TO 15
   22 IF (L2 .EQ. 0) GO TO 23
      J=TI
      GO TO 3
C     * GO TO CHECK THE LEFT HAND END OF A LINE *
   23 TI=J
   24 J=J-1
      IF ( B(I,J) .EQ. 1) GO TO 24
      J1=J+1
      J2=J+2
      J3=J+3
      J4=J+4
      L2=1
      I1=I+1
      I3=I+2
      GO TO 16
    1 CONTINUE
C     ***PRELIMINARY CLASSIFICATION***
      DO 30 J=3,22
      SS=0
      DO 31 I=3,22
   31 SS=SS+B(I,J)
      JJ=J-2
   30 OOC(JJ)=SS
      WRITE (61,44) (OOC(J),J=1,20)
   44 FORMAT (1H ,≠OOC=≠,20(I2,1X))
      CALL MO
      IF ( IO-3) 66,55,66
   55 DO 54 I=3,22
      SS=0
      DO 53 J=3,22
   53 SS=SS+B(I,J)
      II=I-2
   54 OOC(II)=SS
      WRITE (61,44) (OOC(J),J=1,20)
      CALL MO
      IO=3
   66 CALL FC
      CALL OOC
      S=0
      CALL SEARCH
      IF ( S ) 46,45,46
   46 IO=3
      CALL FC
      CALL OOC
      CALL SRLCT
   45 GO TO 999
  101 STOP
      END
```

```
      SUBROUTINE FC
      DIMENSION DC(10),DDL(10),MDC(15),ODC(20)
      DIMENSION TLC(20),LLC(25),LDC(20),B(25,25)
      INTEGER B,DC,DDL,LDC,MDC,ODC,D1,ID
      INTEGER TLC,LLC,S
      COMMON DC,DDL,MDC,D1,ID,MIN2,S,ODC,LDC,B
      ND=ID
C     *STORE ALL LINES IN EACH COLUMN IN TLC *
      IF ( ND .EQ. 3 ) GO TO 1
      JS=0
      GO TO 2
 1    JS=1
 2    DO 3 J=3,22
      L=0
      I=2
 5    I=I+1
      IF ( I .GE. 23) GO TO 11
      IF (JS) 17,7,17
 17   IF ( B(J,I)) 6,5,6
 7    IF (B(I,J)) 6,5,6
 6    K1=0
 8    K1=K1+1
      I=I+1
      IF ( I .GE. 23 ) GO TO 11
      IF (JS) 77,78,77
 77   IF ( B(J,I)) 8,9,8
 78   IF ( B(I,J)) 8,9,8
 9    L=L+1
      TLC(L)=K1
      GO TO 5
 11   MAX=0
      I=0
 15   I=I+1
      IF ( I-L) 12,12,13
 12   IF ( MAX-TLC(I)) 14,15,15
 14   MAX=TLC(I)
      GO TO 15
 13   LLC(J)=MAX
 3    CONTINUE
 61   J=J-1
      IF ( LLC(J)) 62,61,62
 62   NL=J
      WRITE (61,60) (LLC(JJ),JJ=3,J)
 60   FORMAT (1H ,≠LLC≠,20(I2,2X))
C     ***CALCULATE LDC AND DC ***
      DO 72 I=1,4
 72   DC(I)=0
      D1=0
      I=3
      J=I+1
      IF ( LLC(I)-LLC(J)) 21,21,20
 21   I=I+1
      IF ( I .GE. NL) GO TO 33
      J=J+1
      IF ( LLC(I)-LLC(J)) 21,21,20
 20   D1=D1+1
      D2=0
      LDC(D1)=LLC(I)
 25   I=I+1
      J=I+1
      IF (I-NL) 22,33,33
 22   IF ( LLC(I)-LLC(J)) 24,23,23
 23   D2=D2+1
      GO TO 25
 24   D2=D2+1
      I=I+1
      J=I+1
      IF (I-NL) 26,31,31
 26   IF ( LLC(I)-LLC(J)) 24,24,27
 27   DC(D1)=D2
      GO TO 20
 31   DC(D1)=D2
      D1=D1+1
      LDC(D1)=LLC(I)
 33   IA=D1-1
      WRITE (61,70) ( DC(I),I=1,IA)
 70   FORMAT (1H ,≠DC=≠,5(I2,2X))
      WRITE (61,71) (LDC(J),J=1,D1)
 71   FORMAT (1H ,≠LDC=≠,10(I2,2X))
      RETURN
      END
```

```
      SUBROUTINE DOC
      DIMENSION DC(10),DOL(10),MDC(15),DDC(20)
      DIMENSION LDC(20),B(25,25)
      INTEGER B,DC,DOL,LDC,MDC,DDC,D1,S
      COMMON DC,DOL,MDC,D1,ID,MIN2,S,DDC,LDC,B
      DO 1 I=1,10
1     DOL(I)=0
      K1=D1+1
      LDC(K1)=-1
      DO 2 J=1,D1
      MIN=20
      K=0
      DO 3 I=1,D1
      IF ( DOL(I) .NE. 0 ) GO TO 3
      IF ( LDC(I) .GE. MIN ) GO TO 3
      K=I
      MIN=LDC(I)
3     CONTINUE
      IL=MIN-LDC(K1)
      IF (1 .GE. IL ) GO TO 5
      DOL(K)=J
      IF=0
      GO TO 6
5     LDC(K)=LDC(K1)
      IF=IF+1
      DOL(K)=J-IF
6     K1=K
2     CONTINUE
      WRITE (61,30) (DOL(I),I=1,D1)
30    FORMAT (1H ,≠DOL=≠,10(I2,1X))
      RETURN
      END
```

```
      SUBROUTINE MD
      DIMENSION DC(10),DOL(10),MDC(15),DDC(20)
      DIMENSION LDC(20),B(25,25)
      INTEGER B,DC,DOL,LDC,MDC,DDC,D1,S,FL
      COMMON DC,DOL,MDC,D1,ID,MIN2,S,DDC,LDC,B
      DO 50 I=1,15
50    MDC(I)=0
      J=0
1     J=J+1
      IF ( DDC(J) .NE. 0) GO TO 2
      IF ( 20-J) 40,40,1
2     J1=21
4     J1=J1-1
      IF ( DDC(J1) .EQ. 0) GO TO 4
      ID=0
15    I=J+1
      IF ( DDC(J)-DDC(I) ) 12,14,5
14    J=J+1
      GO TO 15
12    FL=0
      GO TO 6
5     FL=1
6     ID=ID+1
      MDC(ID)=DDC(J)
8     J=J+1
      IF ( J .GE. J1 ) GO TO 10
      I=J+1
      IF ( FL .EQ. 0) GO TO 7
      IF  (DDC(J) .GE. DDC(I)) GO TO 8
      GO TO 9
7     IF ( DDC(J) - DDC(I)) 8,8,9
9     ID=ID+1
      MDC(ID)=DDC(J)
13    J=J+1
      IF ( J .GE. J1) GO TO 10
      I=J+1
      IF (FL) 17,11,17
11    IF ( DDC(J)-DDC(I)) 6,13,13
17    IF ( DDC(J)-DDC(I)) 13,13,6
10    ID=ID+1
      MDC(ID)=DDC(J)
      WRITE (61,60) ( MDC(I),I=1,ID)
60    FORMAT (1H , ≠MDC=≠,10I2)
40    RETURN
      END
```

```
         SEARCH
         ENTRY    SEARCH,SRLCT
         EXT      MT
ERM      BCO      4, INPUT ERROR


TRB      BCD      1,
TRA      BCO      2,

OCL      BSS      2
OOLL     BSS      2
IN       BSS      1
NO       BSS      1
LCT      OCT      04000000,01020000,00000000

         OCT      01040000,02010300,00000002

TRAA     BCO      2,STICK

         BCO      2, OU

IT       BSS      1
TA       BSS      2
NOT      OCT      00000000,00000000

         OCT      00000000,01000000

         OCT      00000000,03000002

         OCT      00000000,03000010

         OCT      01000016,01000020

         OCT      03000022,02000030

         OCT      01000034
OCT      OCT      07000000,01000000

         OCT      02000000,01000002

         OCT      05000000,01000004

         OCT      06000000,01000006

         OCT      04040000,01000010

         OCT      05010000,01000012

         OCT      01010500,01000014

         OCT      03010400,01000016
```

```
         SEARCH
         OCT      02030600,01000020

         OCT      02040400,01000022

         OCT      02030303,01000024

         OCT      04030102,01000026

         OCT      03020202,01000030

         OCT      02040200,01000032

         OCT      04010201,01000034

OOLT     OCT      01020000,01000000

         OCT      02010000,01000003

         OCT      01020000,01000006

         OCT      01010000,01000011

         OCT      02020100,01000014

         OCT      02010200,01000017

         OCT      02010204,01000022

         OCT      01030203,01000025

         OCT      02030301,01000030

         OCT      02020201,01000033

         OCT      01040301,01000036

         OCT      02020502,01000041

         OCT      04010104,01000044

         OCT      01040203,01000047

         OCT      01040106,01000052

         COMMON
OC       BSS      10
OOL      BSS      10
MOC      BSS      15
O1       BSS      1
ID       BSS      1
MIN2     BSS      1
S        BSS      1
         PRG
SEARCH   UJP      **
```

```
         SEARCH
         LDA       ID
         STA       ND
         ENI       3,2
         ENI       0,1
PICK1    LDQ       OC,1
         SHQ       18
         SHAQ      6
         INI       1,1
         IJD       PICK1,2
         STA       DCL       STORE 1ST 4 WORD OF
         ENI       3,2       OC IN DCL
         ENI       0,1
PICK2    LDQ       DOL,1
         SHQ       18
         SHAQ      6
         INI       1,1
         IJD       PICK2,2
         STA       DOLL      STORE 1ST 4 WORD OF
         LDI       ND,1      DOL IN DOLL
         LDA       NDT,1
         SWA       IN
         SHA       -18
         TAI       2         STORE NUMBER OF ITEMS
         STI       IT,2
         LDI       IN,1
CNI      LDA       OCT,1     PICK UP AN ITEM IN OCT
         LDQ       DCL
         AQJ,EQ    LOOL      LOOK FOR OOL TABLE
         INI       2,1
         IJD       CNI,2     COMPARE NEXT ITEM
         ENA       ERM
         ENQ       4
         WRITE     61
         UJP       SEARCH
LOOL     INI       1,1
         LDA       OCT,1     PICK UP THE ENTRANCE
         SWA       IN        ADDRESS OF OOLT
         SHA       -18
         TAI       2
         STI       IT,2
         LDI       IN,1
CNIT     LDA       OOLT,1
         LDQ       OOLL
         AQJ,EQ    LMOC      LOOK FOR MOC TABLE
         INI       2,1
         IJD       CNIT,2
         ENA       ERM
         ENQ       4
         WRITE     61
         UJP       SEARCH
LMOC     INI       1,1
         LDA       OOLT,1
         RTJ       MT
         UJP       SEARCH
```

```
         SEARCH
SRLCT    UJP       **
         ENI       3,2
         ENI       0,1
PICKA    LDQ       DC,1
         SHQ       18
         SHAQ      6
         INI       1,1
         IJD       PICKA,2
         STA       DCL
         ENI       3,2
         ENI       0,1
PICKB    LDQ       DDL,1
         SHQ       18
         SHAQ      6
         INI       1,1
         IJD       PICKB,2
         STA       DOLL
         LDI       MIN2,1
         LDA       LCT,1
         LDQ       DCL
         AQJ,NE    NE1
         INI       1,1
         LDA       LCT,1
         LDQ       DOLL
         AQJ,NE    NE2
         INI       1,1
         LDA       LCT,1
         UJP       *+4
NE1      INI       1,1
NE2      INI       1,2
         LDA       LCT,1
         TAI       1
         LDAQ      TRAA,1
         STAQ      TRA
         ENA       TRB
         ENQ       3
         WRITE     61
         UJP       SRLCT
         END

        NUMBER OF LINES WITH DIAGNOSTICS          0
```

```
          MT
          ENTRY     MT
ERN       BCD       4, INPUT ERROR


OCL       BSS       1
OOLL      BSS       1
NO        OCT       00000007
IN        BSS       1
IX        BSS       1
MTL       BSS       1
IT        BSS       1
TA        BSS       2
SSR       BSS       20
IS        BSS       1
SI        BSS       1
J         BSS       1
INW       BSS       1
II        BSS       1
IY        BSS       1
MIN1      BSS       1
TWO       DEC       2
FOUR      DEC       4
TRB       BCD       1,
TRA       BCD       2,

ABS       BSS       1
MOCT      OCT       01130117,00000000,00000000

          OCT       01170305,01000000,00000002

          OCT       01120117,01000000,00000004

          OCT       01160217,01000000,00000006

          OCT       02130213,03100100,00000010

          OCT       02150210,04070100,00000012

          OCT       01100307,04070200,00000014

          OCT       04050116,05110314,00000016

          OCT       11011702,17020301,00000020

          MT
          OCT       04110611,07110611,00000022

          OCT       03060220,02110405,00000024

          OCT       03130215,04120704,00000026

          OCT       01161112,04110517,00000030

          OCT       03050221,03050310,00000032

          OCT       04071704,05021203,00000034

TRAT      BCD       2,LAND

          BCD       2,BALAS

          BCD       2,STOP

          BCD       2,COMPARE

          BCD       2,HAPPY

          BCD       2,CURRENCE

          BCD       2,WANT

          BCD       2,MISS

          BCD       2,HERE

          BCD       2,AGE

          BCD       2,THING

          BCD       2,NOW

          BCD       2,SPECIAL

          BCD       2,SHAPE

          COMMON
DC        BSS       10
DOL       BSS       10
MOC       BSS       15
O1        BSS       1
ID        BSS       1
MIN2      BSS       1
S         BSS       1
```

```
         MT
         PRG
MT       UJP     **
         SWA     IN
         SHA     -18
         TAI     2
         LDA     IN
         STA     INW
         MUA     FOUR
         TAI     1        STORE CHAR ADD IN B1
         STI     IT,1
         LDQ     MDC
NI       LACH    MDCT,1
         AQJ,EQ  BM       BEGIN TO MATCH EACH CHAR
         INI     12,1
         IJD     NI,2
         ENA     ERM
         ENQ     4
         WRITE   61
         UJP     MT
BM       STI     IX,1
         STI     IY,1     CHAR ADDRE OF INITIA
         TIA     1        COMPARE LOCATION
         SHAQ    -24
         DVA     FOUR
         STA     II       WORD ADDRE OF THE INI-
         ENA     0        TIAL COMPARE LOCATION
         STA     SI
         STA     J
RP1      ENA     0
         STA     IS
         LDI     ND,2
         ENI     0,3
RSB      LACH    MDCT,1   COMPARE EACH CHARACTER
         SBA     MDC,3
         STA     ABS
         AQJ,GE  *+2
         LCA     ABS
         ASG     3
         UJP     *+2
         UJP     CMT
         STA     MTL
         MUA     MTL
         RAD     IS       LEAST SQUARE
         INI     1,3
         INI     1,1
         IJD     RSB,2
         LDI     J,3
         LDA     IS
         STA     SSR,3    STORE R**2 IN SSR
         INI     1,3
         LDI     II,1
         INI     2,1
         LDA     MDCT,1
         STA     SSR,3
         MT
         INI     1,3
         STI     J,3
CMT      LDA     SI
         ASE     0
         UJP     UW
         LDI     II,1
         INI     3,1
         STI     II,1
         LDI     IY,1
         INI     12,1
         STI     IY,1     INCREASE THREE WORDS
         LACH    MDCT,1
         SBA     MDC
         STA     ABS
         AZJ,GE  *+2
         LCA     ABS
         ASG     3
         UJP     RP1
         ENA     1
         STA     SI
         LDA     IX
         STA     IY
         LDA     INW
         STA     II       GOES TO UPWARD
UW       LDI     II,1
         INI     -3,1
         STI     II,1
         LDI     IY,1
         INI     -12,1
         STI     IY,1
         LACH    MDCT,1   LOAD THE 1ST CHAR OF
         SBA     MDC      THE WORD BEF THE ENTR-
         STA     ABS      ANCE LOCATION
         AZJ,GE  *+2
         LCA     ABS
         ASG     3
         UJP     RP1
         ENA     20
         STA     MIN1
         ENI     0,1
         ENA,S   -2
         RAD     J
         LDA     J
         AZJ,GE  *+5
         ENA     ERM
         ENQ     4
         WRITE   61
         UJP     MT
         DVA     TWO
         TAI     3
         LDA     MIN1
RM       LDQ     SSR,1
         AQJ,NE  *+5
         ENA     1
         MT
         STA     S
         INI     2,1
         UJP     SED
         AQJ,LT  ALTQ
         ENA     0
         STA     S
         STQ     MIN1
         INI     1,1
         LDQ     SSR,1
         STQ     MIN2
         INI     1,1
         UJP     *+2
ALTQ     INI     2,1
SED      IJD     RM,3     RECOMPARE MINIMUM
         LDA     S
         ASE     1
         UJP     *+2
         UJP     MT
         LDI     MIN2,1
         LDAQ    TRAT,1
         STAQ    TRA
         ENA     TRB
         ENQ     3
         WRITE   61
         UJP     MT
         END

NUMBER OF LINES WITH DIAGNOSTICS     0
```