N71-17338
NASA TM X-2172

NASA TECHNICAL MEMORANDUM

NASA TM X-2172

FORTRAN PROGRAM FOR COMPUTATION OF GROUP TABLE-ALPHANUMERIC DISPLAY

by Gabriel Allen and David D. Evans

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • FEBRUARY 1971

1. Report No. NASA TM X-2172	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle FORTRAN PROGRAM FOR COMPUTATION OF GROUP TABLES - ALPHANUMERIC DISPLAY		5. Report Date February 1971 6. Performing Organization Code
7. Author(s) Gabriel Allen and David D. Evans		8. Performing Organization Report No. E-5927
9. Performing Organization Name and Address Lewis Research Center National Aeronautics and Space Administration Cleveland, Ohio 44135		10. Work Unit No. 129-03 11. Contract or Grant No. 13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, D.C. 20546		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

An existing FORTRAN program for computation of group tables has been modified to allow the output to be displayed as symbols. The procedure for using the program is as follows: An isomorphism is found between the group of interest G and some subgroup T_j of the symmetric group S_n. The elements of T_j are entered as input data in cycle notation. The symbols for the elements of G as they are to appear in the table are also read in. The group multiplication table for those elements of G entered is computed and printed using these symbols.

17. Key Words (Suggested by Author(s))		18. Distribution Statement
Group Machine computation Permutation subgroup	Mathematics Isomorphism	Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	27	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151

FORTRAN PROGRAM FOR COMPUTATION OF GROUP TABLES - ALPHANUMERIC DISPLAY

by Gabriel Allen and David D. Evans

Lewis Research Center

SUMMARY

An existing FORTRAN program for machine computation of finite groups has been modified to allow the output to be displayed as symbols rather than as numerical cycles. The program is written for second generation machines. As written, the program makes use of, but does not depend on, the fact that every finite group G of order n is isomorphic to some subgroup T_j of the symmetric group S_n. The program can print a table of any possible combination of subsets of group elements of G. The procedure for using the program is as follows: After finding the T_j, those elements of T_j which correspond to the subsets of G whose table is desired are read in as input data in cycle notation. The symbol for each element of G as it is to appear in the table is also read in. The group multiplication table for the subsets of G is computed and printed using these symbols. Examples are shown of complete group tables, tables of the group multiplication of subsets and a change in symbol notation.

INTRODUCTION

Since its first timid beginnings sixty years ago, group theory has become a standard branch of applied mathematics. Reference 1, which was published in 1959, has a list of 213 references of applications of group theory. This list (which is not exhaustive) includes applications to crystallography, chemistry, atomic and molecular spectroscopy, elementary particles, and relativity. However, the computational effort involved in dealing with large finite groups can be prohibitive. As a result, computing machines have been applied to the task of proving theorems and algorithms about finite groups whose group tables were entered into the program as input (ref. 2).

Recently, a FORTRAN program was written for the machine computation of the group table itself, using only the elements of the group as input data (ref. 3). This program was also made available for use on second generation machines (ref. 4). The method made use of the fact that every finite group of order n is isomorphic to some subgroup of the symmetric group S_n (ref. 1, p. 260). Therefore, in using this program, the input data were entered in the form of cycles, the elements of S_n. The output was also printed in the form of cycles and the isomorphism between the elements of G and S_n was then used to construct the desired table for G. This last step had to be performed manually.

This program saved the bulk of the effort required to construct a group table. Nevertheless, the fact that the entries in the printed output appeared in the form of cycles meant that a decoding process (the isomorphism between G and the appropriate subgroup of S_n) had to be used to write the final table for G. This decoding process could still be time consuming. The modified program (which is written for second generation machines) performs this decoding process and prints the group table in easily recognized alphanumeric symbols rather than numerical cycles.

The major new function of the program is contained in a subroutine with two entry points. This subroutine identifies a given element of T_j and substitutes for it the symbol for the appropriate element of G in the output format.

With the use of this program, a desired group multiplication table can be obtained in the following manner:

(1) A set of convenient alphanumeric symbols for the elements of G is selected.

(2) An isomorphism between G and some subgroup T_j of S_j is found. Here S_j is the smallest symmetric group which can serve this purpose.

(3) The elements of T_j which correspond to the elements in the subsets of G whose multiplication table is desired are entered as input in the form of cycles.

(4) The symbols chosen in step 1 are entered as input.

(5) The group multiplication table is computed and the output printed using the symbols chosen in step 1.

An example of the group multiplication table of two subsets of a group is presented in the output.

SYMBOLS

A_n	alternating subgroup of symmetric group, S_n
$C_{k,l}$	element in k^{th} row and l^{th} column of matrix C
D_3	dihedral group
G	finite group of order n

O	the cubic group
S_n	symmetric group of order n-factorial
T_j	some subgroup of S_n which is isomorphic to O
ρ_i	twofold rotation axes

DEFINITIONS, CONVENTIONS AND PREPROGRAM PREPARATIONS

The symbol S_n designates the permutation group of n objects and is of order n factorial. The even permutation group of n objects A_n is a proper subgroup of S_n. The numbers between commas in a cycle will be called units. The convention adopted herein for describing the effect of a cycle is that in which the units denote objects and each unit is moved to the location currently occupied by the unit to its left.

Before using the program, some suitable alphanumeric symbols for the elements of G are selected. These symbols would preferably be as close to the symbols in the literature for that group as the printer is capable of printing. Next, an isomorphism is found between G and T_j, the subgroup of some symmetric group S_j, which is as small as can be thought of for the purpose. Finally, the particular sequence of units in a cycle for a given element of S_j must be decided upon, for the main program must store the cycles in this exact form in the doubly dimensioned array representing the group table. See appendix A for an example of how this can be done for S_4.

The user normally has the freedom to choose any sequence of units for an element of S_j which forms a cycle equivalent to the one used herein (see ref. 3). However, in this program the smallest integer in the entire cycle must always be the left-most one.

It should also be noted that the largest group which this program will handle is S_4. Once the principles of the particular decoding process used in this report are understood by the user, the extension to larger groups should be straightforward. However, such an extension will require that subroutine READ be rewritten.

PROGRAM DESCRIPTION

General Description

After the group multiplication has been performed, the results are stored in the doubly dimensioned array C. An entry in this array C_{kl} corresponds to the element in the k^{th} row and the l^{th} column of the table. However, in this array, the element is stored as a cycle chain. The program identifies C_{kl} as a specific member of S_n and prints in its place that alphanumeric symbol chosen to represent the element of G corresponding to the identified element of S_n. This process will be clarified subsequently by an example.

The program as presented in this report makes use of the symmetric group to obtain a group table for T_j which remains internal to the machine. The decoding subroutine can be applied to the translation of any list of cycles as input to any list of desired symbols. All that is necessary is a correspondence between the desired symbols and the cycles. As a matter of fact, some of the complications of the program in reference 4 were needed merely to simplify the output format. In this report, those parts of the program have been eliminated, resulting in a simplification.

Those parts of the program needed for understanding its use are described next. For more detailed information, reference may be made to the listing in appendix B and to the flow chart of subroutine READ in figure 1. The examples are contained in a separate section following this main section.

Main Program

Block 1 - Set up constants. - The constants used in the program are given literal names and are declared either INTEGER or LOGICAL in TYPE statements. All the subscripted variables are dimensioned and alloted storage locations by the use of DIMENSION statements. The names of the variables of the group elements of S_4 are equivalenced to members of the dimensioned variable INPT. A DATA statement is used to give literal names to the following variables:

Variable	Literal name	Identification
(	LP	Left parenthesis
)	RP	Right parenthesis
,	CM	Comma
.	PER	Period
	BLK	Blank
E	IDNT	Identity element of group

Block 2 - Read in data. - The data describing the group, the group elements, and the decoding process are read into the program. The group itself is defined by the variable IDENT. The assignment of labels to the objects and to the locations are read in from the list for IDENT which is limited to one card. No shifting or 'SQUEZ'ing is required, thus simplifying the programming compared to reference 4. The input cards are read in the following order:

First card: TITLE, one card with FORMAT (80A1)

Second card: IDENT, one card with FORMAT (80A1)

The input for IDENT must be long enough to include the maximum number of objects to be moved in any group operation. Thus, the list for IDENT must be of the form(A_1, A_2, . . . , A_j) where j is the subscript of S_j and A_i is any alphanumeric symbol.

Several error checks are run on IDENT. These include checking the maximum number of nonblank units (which is six in this version), the presence of blank spaces on the card, and the illegal use of parentheses.

The set of group elements whose multiplication table is desired (this may be a subset of the entire group) is read in the next two sets of cards. These two sets are identified in the program as INPUT(J,K). The first of these sets contains the ACROSS (column) operations and consists of the INPUT cards for which K = 1. There can be as many as four cards with FORMAT(80A1), and each operation is followed by a period.

A blank card follows the preceding set. It is needed to signal the end of the ACROSS input.

The next set contains the DOWN (row) operations and consists of INPUT cards for which K = 2. There can be as many as four cards with FORMAT(80A1), and again, each operation is followed by a period.

A blank card follows the preceding set, signaling the end of the DOWN input.

The input for both ACROSS and DOWN operations is in cycle notation with the smallest integer consistently to the left. No blanks are allowed between the first left parenthesis and the final right parenthesis in any cycle chain. If integers are not used as the units in the cycle, subroutine READ will have to be rewritten to identify the units which are used. The complete set of column operations is read first. Then the complete set of row operations is read. Each of these sets is limited to four cards and is followed by a blank card.

The FORMAT for the input list is A6. All entries must be left-adjusted in order that the output be in line. The example of input which follows is a class product block from S_4 (see ref. 5). It is the product of the class R1, R2, R3 (see appendix A) times the class containing the rotations about all the threefold axes.

```
Card
 1   S4
 2   (1,2,3,4)
 3   (2,4,3).(1,2,3).(1,4,2).(1,3,4).(2,3,4).(1,3,2).(1,2,4).(1,4,3).
 4
 5   (1,3)(2,4).(1,2)(3,4).(1,4)(2,3).
 6
 7   I1    I2    I3    I4    I5    I6    SR1   SR2   SR3   SR4   SR1S  SR2S
 8   SR3S  SR4S  R1    R2    R3    R1S   R2S   R3S   R1C   R2C   R3C
```

Card 3 contains the ACROSS (column) operations, card 5 contains the DOWN (row) operations, and cards 7 and 8 contain the list of "values" for INPT. Note that the entire group is listed even though only two classes are multiplied. This is done because classes are not closed with respect to group multiplication and elements of S_4 which are not in either of the classes used as input can appear in the product.

See reference 4 for a description of the rest of the main program.

The data for the decoding process are read from the list for the variable INPT. The list for INPT consists of the actual names desired for the group elements in the output. The list, of course, assigns "values" to the dimensioned variable INPT. The "variable" names are listed in an equivalence statement in block 1. In this program, the alphanumeric data in the list for INPT (the values) are the same as the symbols in the equivalence statement (the variables). For example, the 11^{th} value on the list for INPT is SR1S; and in block 1, INPT(11) is equivalenced to the variable SR1S.

This procedure makes it convenient to use any other choice of variables which are isomorphic to the group in place of the ones used here. If the symbol Q5 were desired in place of SR1S, assigning the value Q5 to the variable SR1S would cause Q5 to be printed everywhere that SR1S appears now. The important point is that no change in the body of the program itself is required. The single change in the input list takes care of the entire new decoding process.

An example of such a substitution is shown in the output for S_3. In the example, the well-known isomorphism between S_3 and the dihedral group D_3 is used to provide symbols for the elements of S_3 which are more covenient than the ones which arise when S_3 is regarded as a subgroup of S_4. The common symbols for the elements of D_3 are used. These are a threefold rotation axis R and three twofold rotation axes ρ_1, ρ_2, and ρ_3. This new output was obtained by simply changing cards 7 and 8 to read as follows:

```
|RHO1 | |I2    I3    RHO3  I5    RHO2  SR1   R     SR3   SR4   SR1S  RS    |
|SR3S | |SR4S  R1    R2    R3    R1S   R2S   R3S   R1C   R2C   R3C         |
```

Subroutine READ

This is the part of the program which decodes the cycles stored in INPUT and OUT (see listing) into the desired symbols for the elements of G. A process of elimination is used to identify the cycle, character by character. Advantage is taken of the fact that all equivalent cycles are stored in the same unique form (see ref. 3). Each integer in a given cycle is identified in succession beginning with the left-most one. The identification of an integer is a simple comparison process between the values assigned to the variables "ONE," "TWO," "THREE" and the unit of the cycle. When needed, left and right parentheses are identified by comparing them against the values of the variables LP and RP, respectively. After a given character has been identified, control is shifted to another set of comparison tests to be made on the next character in the cycle. As each character of the cycle is identified, the number of possible group elements to which the cycle could correspond decreases. The identification is complete when there is only one group element which could correspond to the given cycle.

To keep the program as short as possible, it is assumed that certain errors will not be made. For example, if 4 is the largest integer to appear in a cycle, tests are made only up to and including the integer 3. If the number in the cycle fails all of the tests, it is assumed that it is a 4 in the identifying process.

Illustrative example 1 - Identification of the cycle (1, 3, 4, 2) as the group element R_3. - The cycle (1, 3, 4, 2) enters the subroutine as the value of the variable INPUT in the calling vector. The identification proceeds by examining individual characters in the array in sequence. This array has four integers, three commas, a left parenthesis, and a right parenthesis, for a total of nine characters. The first character to be examined is the left-most one. Since it is a left parenthesis, it fails all the comparison tests and the next character is examined. This one passes the comparison test on the integer 1 and control is thereby shifted to statement 1100. The comparison test in this series of statements takes into account the fact that some unit has been identified before transfer to statement 1100. Since the first integer in any cycle is always followed by a comma, the subroutine skips the next entry and goes to entry I + 2 for the next comparison test. Since this entry is the integer 3, control is shifted to statement 1130.

The $(I + 3)^{rd}$ entry in the array is examined next. It is not a right parenthesis, so control is shifted to statement 1140. The $(I + 4)^{th}$ and $(I + 5)^{th}$ entries both fail their comparison tests and the array is thus uniquely identified as the group element R_3.

Illustrative example 2 - Identification of the cycle chain (1, 3)(2, 4) as the element R1S. - This particular array has a total of 10 characters divided amongst four integers, and two each of commas, and left and right parentheses. The first entry in the array (the K^{th} one) is again a left parenthesis. After it has been identified, the $(K + 2)^{nd}$ entry (K + 2) is examined. Since it is a 1, control is shifted to statement 1100. As in the previous example, the next entry to be examined is the $(K + 2)^{nd}$ one. Since it is a 3, control is shifted to statement 1130. After the comparison tests in this series are completed, control is shifted to statement 1131, where the identification R1S is made.

EXAMPLES OF OUTPUT

Full Group

```
       S4
  (1,2,3,4)
(1,2).(3,4).(1,4).(2,3).(2,4).(1,3).(2,4,3).(1,2,3).(1,4,2).(1,3,4).
(2,3,4).(1,3,2).(1,2,4).(1,4,3).(1,2,3,4).(1,4,2,3).(1,3,4,2).(1,3)(2,4).
(1,2)(3,4).(1,4)(2,3).(1,4,3,2).(1,3,2,4).(1,2,4,3).

(1,2).(3,4).(1,4).(2,3).(2,4).(1,3).(2,4,3).(1,2,3).(1,4,2).(1,3,4).
(2,3,4).(1,3,2).(1,2,4).(1,4,3).(1,2,3,4).(1,4,2,3).(1,3,4,2).(1,3)(2,4).
(1,2)(3,4).(1,4)(2,3).(1,4,3,2).(1,3,2,4).(1,2,4,3).
```

S4

	I1	I2	I3	I4	I5	I6	SR1	SR2	SR3	SR4	SR1S	SR2S	SR3S	SR4S	R1	R2	R3	R1S	R2S	R3S
I1	E	R2S	SR3	SR2	SR3S	SR2S	R3C	I4	I3	R3	R1	I6	I5	R1C	SR1S	R3S	SR4	R2C	I2	R2
I2	R2S	E	SR4	SR1	SR1S	SR4S	I4	R3C	R3	I3	I5	R1C	R1	I6	SR3S	R1S	SR3	R2	I1	R2C
I3	SR3S	SR4S	E	R3S	SR3	SR4	R1C	R1	I5	I6	R2	R2C	I1	I2	SR2	SR1S	R1S	R3	R3C	I4
I4	SR2S	SR1S	R3S	E	SR1	SR2	I5	I6	R1C	R1	I2	I1	R2C	R2	SR4	SR4S	R2S	R3C	R3	I3
I5	SR3	SR1	SR3S	SR1S	E	R1S	I2	R2	I1	R2C	I4	R3	I3	R3C	R3S	SR2	SR2S	I6	R1C	R1
I6	SR2	SR4	SR4S	SR2S	R1S	E	R2C	I1	R2	I2	R3	I4	R3C	I3	R2S	SR3	SR1S	I5	R1	R1C
SR1	R1C	I5	R2C	I2	I4	R3C	SR1S	SR4S	SR2S	SR3S	E	R2S	R3S	R1S	I3	I6	I1	SR2	SR3	SR4
SR2	I6	R1	R2	I1	R3C	I4	SR3S	SR2S	SR4S	SR1S	R2S	E	R1S	R3S	R3	R1C	I2	SR1	SR4	SR3
SR3	I5	R10	I1	R2	I3	R3	SR4S	SR1S	SR3S	SR2S	R3S	R1S	E	R2S	I4	R1	R2C	SR4	SR1	SR2
SR4	R1	I6	I2	R2C	R3	I3	SR2S	SR3S	SR1S	SR4S	R1S	R3S	R2S	E	R3C	I5	R2	SR3	SR2	SR1
SR1S	R3	I4	R1	I5	I2	R2	E	R1S	R2S	R3S	SR1	SR3	SR4	SR2	R2C	R3C	R1C	SR4S	SR2S	SR3S
SR2S	I4	R3	R1C	I6	R2C	I1	R1S	E	R3S	R2S	SR4	SR2	SR1	SR3	I2	I3	R1	SR3S	SR1S	SR4S
SR3S	I3	R30	I5	R1	I1	R2C	R2S	R3S	E	R1S	SR2	SR4	SR3	SR1	R2	I4	I6	SR2S	SR4S	SR1S
SR4S	R3C	I3	I6	R1C	R2	I2	R3S	R2S	R1S	E	SR3	SR1	SR2	SR4	I1	R3	I5	SR1S	SR3S	SR2S
R1	SR4	SR2	SR1S	SR3S	R2S	R3S	I1	R2C	I2	R2	R3C	I3	R3	I4	R1S	SR1	SR4S	R1C	I6	I5
R2	R1S	R3S	SR2	SR3	SR4S	SR1S	I3	R3	R3C	I4	R1C	I5	I6	R1	SR2S	R2S	SR1	I2	R2C	I1
R3	SR1S	SR2S	R2S	R1S	SR4	SR3	I6	I5	R1	R1C	R2C	R2	I2	I1	SR1	SR3S	R3S	I3	I4	R3C
R1S	R2	R2C	R3C	R3	I6	I5	SR4	SR3	SR2	SR1	SR2S	SR1S	SR4S	SR3S	R1C	I1	I4	E	R3S	R2S
R2S	I2	I1	R3	R3C	R1	R1C	SR2	SR1	SR4	SR3	SR3S	SR4S	SR1S	SR2S	I5	R2C	I3	R3S	E	R1S
R3S	R2C	R2	I4	I3	R1C	R1	SR3	SR4	SR1	SR2	SR4S	SR3S	SR2S	SR1S	I6	I2	R3C	R2S	R1S	E
R1C	SR1	SR3	SR2S	SR4S	R3S	R2S	R2	I2	R2C	I1	I3	R3C	I4	R3	E	SR4	SR3S	R1	I5	I6
R2C	R3S	R1S	SR1	SR4	SR2S	SR3S	R3	I3	I4	R3C	I6	R1	R1C	I5	SR4S	E	SR2	I1	R2	I2
R3C	SR4S	SR3S	R1S	R2S	SR2	SR1	R1	R1C	I6	I5	I1	I2	R2	R2C	SR3	SR2S	E	I4	I3	R3

	R1C	R20	R3C
I1	SR4S	R1S	SR1
I2	SR2S	R3S	SR2
I3	SR1	SR2S	R2S
I4	SR3	SR3S	R1S
I5	R2S	SR4	SR4S
I6	R3S	SR1	SR3S
SR1	R3	R1	R2
SR2	I3	I5	R2C
SR3	R3C	I6	I2
SR4	I4	R1C	I1
SR1S	I1	I3	I6
SR2S	R2	R3C	I5
SR3S	I2	R3	R1C
SR4S	R2C	I4	R1
R1	E	SR3	SR2S
R2	SR3S	E	SR4
R3	SR2	SR4S	E
R1S	R1	I2	I3
R2S	I6	R2	I4
R3S	I5	I1	R3
R1C	R1S	SR2	SR1S
R2C	SR1S	R2S	SR3
R3C	SR4	SR1S	R3S

Class Algebra Block of S_4

S4
(1,2,3,4)
(2,4,3).(1,2,3).(1,4,2).(1,3,4).(2,3,4).(1,3,2).(1,2,4).(1,4,3).

(1,3)(2,4).(1,2)(3,4).(1,4)(2,3).

S4

	SR1	SR2	SR3	SR4	SR1S	SR2S	SR3S	SR4S
R1S	SR4	SR3	SR2	SR1	SR2S	SR1S	SR4S	SR3S
R2S	SR2	SR1	SR4	SR3	SR3S	SR4S	SR1S	SR2S
R3S	SR3	SR4	SR1	SR2	SR4S	SR3S	SR2S	SR1S

S_3 as a Subgroup of S_4

S3
(1,2,3)
(1,2).(1,3).(2,3).(1,2,3).(1,3,2).

(1,2).(1,3).(2,3).(1,2,3).(1,3,2).

S3

	I1	I6	I4	SR2	SR2S
I1	E	SR2S	SR2	I4	I6
I6	SR2	E	SR2S	I1	I4
I4	SR2S	SR2	E	I6	I1
SR2	I6	I4	I1	SR2S	E
SR2S	I4	I1	I6	E	SR2

S_3 Regarded as Isomorphic to D_3

S3
(1,2,3)
(1,2).(1,3).(2,3).(1,2,3).(1,3,2).

(1,2).(1,3).(2,3).(1,2,3).(1,3,2).

S3

	RHO1	RHO2	RHO3	R	RS
RHO1	E	RS	R	RHO3	RHO2
RHO2	R	E	RS	RHO1	RHO3
RHO3	RS	R	E	RHO2	RHO1
R	RHO2	RHO3	RHO1	RS	E
RS	RHO3	RHO1	RHO2	E	R

01 UNIT05, ECF.   REC= 00000 FIL= 00002

SUMMARY OF PROCEDURE

A FORTRAN program for machine computation of finite groups has been written which allows the output to be displayed in alphanumeric symbols rather than in numerical cycles. The program is written for second generation machines. As written, it makes use of the fact that every finite group G of order n is isomorphic to some subgroup T_j of the symmetric group S_n. However, the method is not dependent on this fact. The program can print a group multiplication table of the product of any two subsets of G.

The main new feature of the program is an efficient decoding process from cycles to any chosen alphanumeric symbols. This decoding process is not limited to elements of groups. It can be applied to the translation of any list of cycles as input to any desired symbols as output.

The procedure for using the program is as follows:

1. A set of desired alphanumeric symbols for the elements of G is selected.
2. An isomorphism between G and some subgroup T_n of S_n is found.
3. The elements in the subsets of G whose group multiplication table is desired are entered as input in the form of cycles.
4. The symbols chosen in step 1 are entered as input.
5. The group multiplication table is computed and the output printed using the symbols chosen in step 1.

The program is complete for groups up to those which are isomorphic to S_4 or any of its subgroups. A straightforward modification can extend it to larger groups if desired.

A change in choice of symbols for elements of G is very easy (see step 4 in the procedure) and an example of such a change is shown. Examples of both a complete group table and a group multiplication table of two subsets of a group are also shown.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, October 7, 1970,
129-03.

APPENDIX A

FORTRAN NOTATION FOR GROUP O FOR CODING OF ISOMORPHISM WITH S_4

In the output examples in this report, the following notation is used: The group O, the cubic group, is isomorphic to the symmetric group S_4. The former group has four threefold axes, three fourfold axes and six twofold axes. Since FORTRAN IV does not have lower-case letters, the following alphanumeric symbols will be used to represent the group elements of O.

The symbol R denotes a rotation about one of the fourfold axes, SR (for small R) represents a rotation about one of the threefold axes, and I represents a rotation about a twofold axis. The particular axis in a given case is identified by a number. The letter S following the axis identification means squared and signifies that the rotation has been repeated about the given axis. The letter C following the axis identification means cubed and signifies that the rotation about the given axis has taken place three times. Thus, R2C is the group element of O signifying that a rotation has taken place three times about fourfold axis number 2; whereas SR1S means that a rotation has occurred twice about threefold rotation axis number 1. (See ref. 5, p. 27, for a convenient table of group O using this notation.)

The following code is used for the isomorphism between cycles (the elements of S_4) and the elements of O:

I1 ↔ (1, 2); I2 ↔ (3, 4); I3 ↔ (1, 4); I4 ↔ (2, 3); I5 ↔ (2, 4); I6 ↔ (1, 3)

SR1 ↔ (2, 4, 3); SR2 ↔ (1, 2, 3); SR3 ↔ (1, 4, 2); SR4 ↔ (1, 3, 4)

SR1S ↔ (2, 3, 4); SR2S ↔ (1, 3, 2); SR3S ↔ (1, 2, 4); SR4S ↔ (1, 4, 3)

R1S ↔ (1, 3)(2, 4); R2S ↔ (1, 2)(3, 4); R3S ↔ (1, 4)(2, 3)

R1 ↔ (1, 2, 3, 4); R2 ↔ (1, 4, 2, 3); R3 ↔ (1, 3, 4, 2)

R1C ↔ (1, 4, 3, 2); R2C ↔ (1, 3, 2, 4); R3C ↔ (1, 2, 4, 3)

APPENDIX B

PROGRAM LISTING

```
C       BLOCK(1) SETS UP CONSTANTS

        DIMENSION ITOP(120),ISID(120),IDOT(22),LINPUT(15),INPT(24)
        DIMENSION INPUT(400,2), INITAL(6), IDENT(80), NPAIR(24,2),
       1LINE(576),NIN(24,2),NOPA(2),KN(6),NINPUT(2)
        INTEGER RP,CM,TEMP,BLK,CHAR,PER,PAIRS(18,24,2),ANS(6,24,24),OUT(15
       1,24,24),TITLE(80),OUTPUT(21)
        INTEGER SR1,SR2,SR3,SR4,SR1S,SR2S,SR3S,SR4S,R1,R2,R3,R1S,R2S,R3S,
       .R1C,R2C,R3C
        DATA IDIT/6H....../
        DATA(IDOT(I),I=1,21)/21*6H....../
        DATA LP,RP,CM,BLK,PER/1H(,1H),1H,,1H ,1H./
        LOGICAL COMMA,ONE
        COMMON/IKYMAX/IKY,MAX,NCHK
        COMMON/NUMOPS/JO,KO,NOUT(24,24)
        COMMON/LITRAL/II1,II2,I3,I4,I5,I6,SR1,SR2,SR3,SR4,SR1S,SR2S,SR3S,
       .SR4S,R1,R2,R3,R1S,R2S,R3S,R1C,R2C,R3C
        EQUIVALENCE (INPT(1),II1)

C       BLOCK(2) READS IN DATA

1       READ (5,66) TITLE
        READ (5,66) IDENT
        WRITE(6,80) TITLE
        WRITE(6,81) IDENT
        ILP=0
        IRP=0
        N=0
        CHAR=BLK
        DO 5 I=1,80
        TEMP=IDENT(I)
        IF (TEMP.EQ.BLK) GO TO 5
        IF (TEMP.EQ.LP) GO TO 2
        IF (TEMP.EQ.RP) GO TO 3
        IF (TEMP.EQ.CM) GO TO 4
        CHAR=TEMP
        GO TO 5
2       ILP=ILP+1
        GO TO 5
3       IRP=IRP+1
4       N=N+1
        IF (N.GT.6) GO TO 61
        IF (CHAR.EQ.BLK) GO TO 62
        INITAL(N)=CHAR
        CHAR=BLK
5       CONTINUE
        IF (ILP.NE.1.OR.IRP.NE.1) GO TO 63
        ID=80
        DO 8 K=1,2
        DO 7 I=1,5
        JS=(I-1)*80+1
```

```
      JE=I*80
      READ (5,66) (INPUT(J,K),J=JS,JE)
      WRITE(6,90) (INPUT(J,K),J=JS,JE)
      DO 6 J=JS,JE
      IF (INPUT(J,K).NE.BLK) GO TO 7
6     CONTINUE
      NINPUT(K)=JS
      GO TO 8
7     CONTINUE
8     CONTINUE
      JE=JE-80
      READ(5,5678) INPT
 5678 FORMAT(12A6)

C     BLOCK(3) STORES EACH OPERATION IN PAIRS ARRAY AS A PRODUCT OF
C                 TRANSPOSITIONS

      DO 19 K=1,2
      NPAIR(1,K)=0
      NOP=1
      IPRN=0
      CHAR=BLK
      II=0
      ONE=.TRUE.
      COMMA=.FALSE.
      IE=NINPUT(K)
      DO 16 I=1,IE
      TEMP=INPUT(I,K)
      IF (TEMP.EQ.BLK) GO TO 16
      IF (TEMP.EQ.LP) GO TO 9
      IF (TEMP.EQ.RP) GO TO 10
      IF (TEMP.EQ.CM) GO TO 11
      IF (TEMP.EQ.PER) GO TO 14
      CHAR=TEMP
      GO TO 15
9     IPRN=IPRN+1
      ONE=.TRUE.
      IF (IPRN.LT.0.OR.IPRN.GT.1) GO TO 57
      GO TO 15
10    IPRN=IPRN-1
      IF (.NOT.COMMA) GO TO 58
      COMMA=.FALSE.
      ONE=.TRUE.
      IF (IPRN.LT.0.OR.IPRN.GT.1) GO TO 57
      GO TO 12
11    COMMA=.TRUE.
12    NPAIR(NOP,K)=NPAIR(NOP,K)+1
      IF (NPAIR(NOP,K).GT.17) GO TO 59
      IF (CHAR.EQ.BLK) GO TO 56
      IJ=NPAIR(NOP,K)
      PAIRS(IJ,NOP,K)=CHAR
      IF (ONE) GO TO 13
      NPAIR(NOP,K)=NPAIR(NOP,K)+1
      IJ=NPAIR(NOP,K)
      PAIRS(IJ,NOP,K)=CHAR
13    CHAR=BLK
      ONE=.FALSE.
      GO TO 15
```

```
14     NOP=NCP+1
       IF (NOP.GT.24) GO TO 60
       NPAIR(NOP,K)=0
15     II=II+1
       INPUT(II,K)=TEMP
16     CONTINUE
       NINPUT(K)=II+1
       IF (INPUT(II,K).NE.PER) GO TC 17
       NOP=NOP-1
       GO TO 18
17     INPUT(II+1,K)=PER
18     IF (NPAIR(1,K).EQ.0) GO TO 1
       NOPA(K)=NOP
19     CONTINUE

C      BLOCK(4) SETS UP ANS ARRAY IN STANDARD FORM OR CONFIGURATION

       NOP1=NOPA(1)
       NOP2=NOPA(2)
       DO 2C I1=1,NOP1
       DO 20 I2=1,NOP2
       DO 2C I=1,N
20     ANS(I,I1,I2)=INITAL(I)

C      BLCCK(5) PERFORMS THE PRODUCT OPERATIONS

       DO 27 I1=1,NOP1
       K1=NPAIR(I1,1)
       DO 27 I2=1,NOP2
       K2=NPAIR(I2,2)
       DO 23 I=1,K1,2
       J=K1-I+1
       J1=0
       J2=0
       DO 22 L=1,N
       IF (ANS(L,I1,I2).NE.PAIRS(J,I1,1)) GO TO 21
       J1=L
21     IF (ANS(L,I1,I2).NE.PAIRS(J-1,I1,1)) GO TO 22
       J2=L
22     CONTINUE
       IF (J1.EC.0.OR.J2.EQ.0) GO TO 55
       ITMP=ANS(J1,I1,I2)
       ANS(J1,I1,I2)=ANS(J2,I1,I2)
       ANS(J2,I1,I2)=ITMP
23     CONTINUE
       DO 26 I=1,K2,2
       J=K2-I+1
       J1=0
       J2=0
       DO 25 L=1,N
       IF (ANS(L,I1,I2).NE.PAIRS(J,I2,2)) GO TO 24
       J1=L
24     IF (ANS(L,I1,I2).NE.PAIRS(J-1,I2,2)) GO TO 25
       J2=L
25     CONTINUE
       IF (J1.EC.0.OR.J2.EQ.0) GO TC 55
       ITMP=ANS(J1,I1,I2)
       ANS(J1,I1,I2)=ANS(J2,I1,I2)
       ANS(J2,I1,I2)=ITMP
```

```
26    CONTINUE
27    CONTINUE

C     BLOCK(6) IDENTIFIES THE SINGLE GROUP ELEMENT WHICH HAS THE SAME
C              EFFECT AS BLOCK(5)

      MAX=0
      DO 38 I1=1,NOP1
      DO 38 I2=1,NOP2
      JRP=0
      J=1
      K1=1
      KA=1
      OUT(1,I1,I2)=LP
      DO 28 KK=1,N
   28 KN(KK)=KK
      DO 35 I=1,N
      DO 29 L=1,N
      KV=L
      IF(ANS(KA,I1,I2) .EQ. INITAL(KV)) GO TO 30
   29 CONTINUE
   30 KN(KV)=0
      IF(KV .NE. K1) GO TO 34
      DO 32 KK=1,N
      KL=KN(KK)
      IF(KL .NE. 0) GO TO 33
   32 CONTINUE
      IF(OUT(J,I1,I2) .NE. CM) GO TO 360
      OUT(J,I1,I2)=RP
      JRP=J
      GO TO 36
   33 K1=KL
      KA=KL
      GO TO 35
   34 IF(OUT(J,I1,I2).EQ. LP) GO TO 335
      J=J+1
      OUT(J,I1,I2)= ANS(KA,I1,I2)
  532 DO 332 KK=1,N
      KL=KN(KK)
      IF(KL .NE. 0) GO TO 333
  332 CONTINUE
  333 IF(KL .NE. K1) GO TO 334
      KN(KL)=0
      GO TO 532
  334 IF(KL .EQ. 0) GO TO 435
      J=J+1
      OUT(J,I1,I2)=CM
      KA=KV
      GO TO 35
  435 J=J+1
      OUT(J,I1,I2)=RP
      JRP=J
      GO TO 36
  335 J=J+1
      OUT(J,I1,I2)=INITAL(K1)
      J=J+1
      OUT(J,I1,I2)=CM
      J=J+1
      OUT(J,I1,I2)=ANS(KA,I1,I2)
```

```
      IF(ANS(KV,I1,I2) .EQ. INITAL(KA)) GO TO 336
      J=J+1
      OUT(J,I1,I2)=CM
      KA=KV
      GO TO 35
  336 KN(KA)=0
      J=J+1
      OUT(J,I1,I2)=RP
  632 DO 432 KK=1,N
      KL=KN(KK)
      IF(KL .NE. 0) GO TO 433
  432 CONTINUE
  433 IF(KL .NE. K1) GO TO 434
      KN(KL)= 0
      GO TO 632
  434 JRP=J
      IF(KL .EQ. 0) GO TO 36
      J=J+1
      OUT(J,I1,I2)=LP
      K1=KL
      KA=KL
   35 CONTINUE
  360 IF(OUT(J,I1,I2) .NE. LP) GO TO 36
      OUT(J,I1,I2)=BLK
   36 J1=J
      J=JRP
      IF(J.NE.0)GO TO 37
      OUT(J1,I1,I2)=BLK
   37 NOUT(I1,I2)=J
      IF (J.LE.MAX) GO TO 38
      MAX=J
38    CONTINUE

C     BLOCK(7)  DECODES AND WRITES OUTPUT
C
      DO 42 L=1,576
42    LINE(L)=BLK
      IKY=0
      DO 45 L2=1,2
      NCHK=0
      NPER=0
      DO 45 L1=1,400
      IF(NPER .GE. NOPA(L2)) GO TO 45
      IF(INPUT(L1,L2) .EQ. PER) GO TO 445
      NCHK=NCHK+1
      NCHK1=NCHK
      GO TO 45
  445 NL=L1-NCHK-1
      NPER=NPER+1
      DO 43 I=1,NCHK
      NL=NL+1
      LINPUT(I)=INPUT(NL,L2)
   43 CONTINUE
      LTAG=1
      CALL READ(LINPUT,LINE,LTAG)
      NCHK=0
   45 CONTINUE
      DO 101 I=1,NOP1
```

```
  101 ITOP(I)=LINE(I)
      IND=0
      NA= NOP1+1
      NAMAX=NOP1+NOP2
      DO 102 I=NA,NAMAX
      IND=IND+1
  102 ISID(IND)=LINE(I)
      DO 46 L=1,NAMAX
46    LINE(L)=BLK
      IKY=0
      LTAG=0
      NCHK=15
      DO 52 KO=1,NOP2
      DO 52 JO=1,NOP1
      DO 50 IO=1,15
   50 LINPUT(IO)=OUT(IO,JO,KO)
      CALL READ(LINPUT,LINE,LTAG)
   52 CONTINUE
      WRITE (6,68) TITLE
      KSW=0
      LSW=0
      DO 115  IL=1,6
      IL1= IL-1
      IF(KSW.EQ.1) GO TO 115
      MAX= 20*IL
      IF(NOP1 .GT. 119) GO TO 111
      IF(NOP1 .GT. MAX)  GO TO 104
      MAX= NOP1
      KSW=1
      IF(NOP1 .LE. 20) MAX= NOP1
  104 IN=0
      ITEM= 20*IL1+1
      DO 105 IJ=ITEM,MAX
      IN= IN+1
  105 OUTPUT(IN)=ITOP(IJ)
      IN1= IN+1
      DO 1055 IP=IN1,21
      IG=IP+1
      IDOT(IG)=BLK
 1055 OUTPUT(IP)= BLK
 1056 WRITE(6,1104) (OUTPUT(I),I=1,20)
 1104 FORMAT(1H0,10X,20A6)
      WRITE(6,1144) (IDOT(I),I=1,21)
 1144 FORMAT(1H ,3X,21A6)
      DO 1145 IR=1,21
 1145 IDOT(IR)=IDIT
      LB=0
  106 DO 108 LN=1,NOP2
      LG=1
      LANN=LB+LN
      IF(LANN .GT. 119) GO TO 109
      OUTPUT(1)=ISID(LANN)
      DO 107 IJ=1,MAX
      NMAX=20*IL1+IJ
      IF(NMAX .GT. NOP1) GO TO 107
      LG=LG+1
      IF(IL .LT. 2) GO TO 1076
      MAR=20*(IL+LN-2)+(NOP1-20)*(LN-1)+IJ
      GO TO 1077
```

```
 1076 MAR=NOP1*(IL +LN-2)+IJ
 1077 OUTPUT(LG)=LINE(MAR)
  107 CONTINUE
      LG=LG+1
      IF(LG .GT. 21) GO TO 1089
      DO 1C88 IQ = LG,21
 1088 OUTPUT(IQ)=BLK
 1089 CONTINUE
  108 WRITE(6,1105) (OUTPUT(I),I=1,21)
 1105 FORMAT(1H ,4X,A6,20A6)
  109 CONTINUE
      IF(LSW .EQ. 1) GO TO 115
      IF(NCP2 .LE. 56) GO TO 115
      IF(NCP2 .GT. 112) GO TO 110
      LSW=1
      LB=56
      GO TC 106
  110 IF(NCP2 .GT. 120) GO TO 111
      LSW=1
      LB= 112
      GO TC 106
  111 WRITE(6,1106) NOP1, NOP2
 1106 FORMAT(1HK,39HTOO MANY ROW OR COLUMN OPERATIONS NOP1=I4,5HNOP2=I4)
  115 CONTINUE
      GO TC 1
55    WRITE (6,71) (IDENT(I),I=1,ID)
      WRITE(6,79) TITLE
      GO TC 1
56    WRITE (6,72)
      GO TC 1
57    WRITE (6,73)
      GO TC 1
58    WRITE (6,74)
      GO TC 1
59    WRITE (6,75)
      GO TC 1
60    WRITE (6,76)
      GO TC 1
61    WRITE (6,77)
      GO TC 65
62    WRITE (6,72)
      GO TC 65
63    WRITE (6,73)
      GO TC 65
65    RETURN

66    FORMAT (80A1)
   68 FORMAT(1HK,50X,20A1)
   71 FORMAT(1HJ,44HILLEGAL ELEMENT IN GROUP. IDENTITY GROUP IS ,11A6)
   72 FORMAT(1HJ,29HBLANK IS NOT A VALID ELEMENT.)
   73 FORMAT(1HJ,51HILLEGAL USE OF PARENS OR TITLE  + IDENT CARDS MIXED)
   74 FORMAT(1HJ,14HILLEGAL GROUP.)
   75 FORMAT(1HJ,34HTHE PAIRS ARRAY HAS BEEN EXCEEDED.)
   76 FORMAT(1HJ,34HTOO MANY OPERATIONS.  LIMIT IS 24.)
   77 FORMAT(1HJ,39HMORE THAN 6 ELEMENTS IN IDENTITY GROUP.)
   79 FORMAT(1HJ,27HCHECK INPUT CARDS IN GROUP ,80A1)
   80 FORMAT(1H1,80A1)
   81 FORMAT(1HJ,80A1)
   90 FORMAT(1HJ,80A1)
      END
```

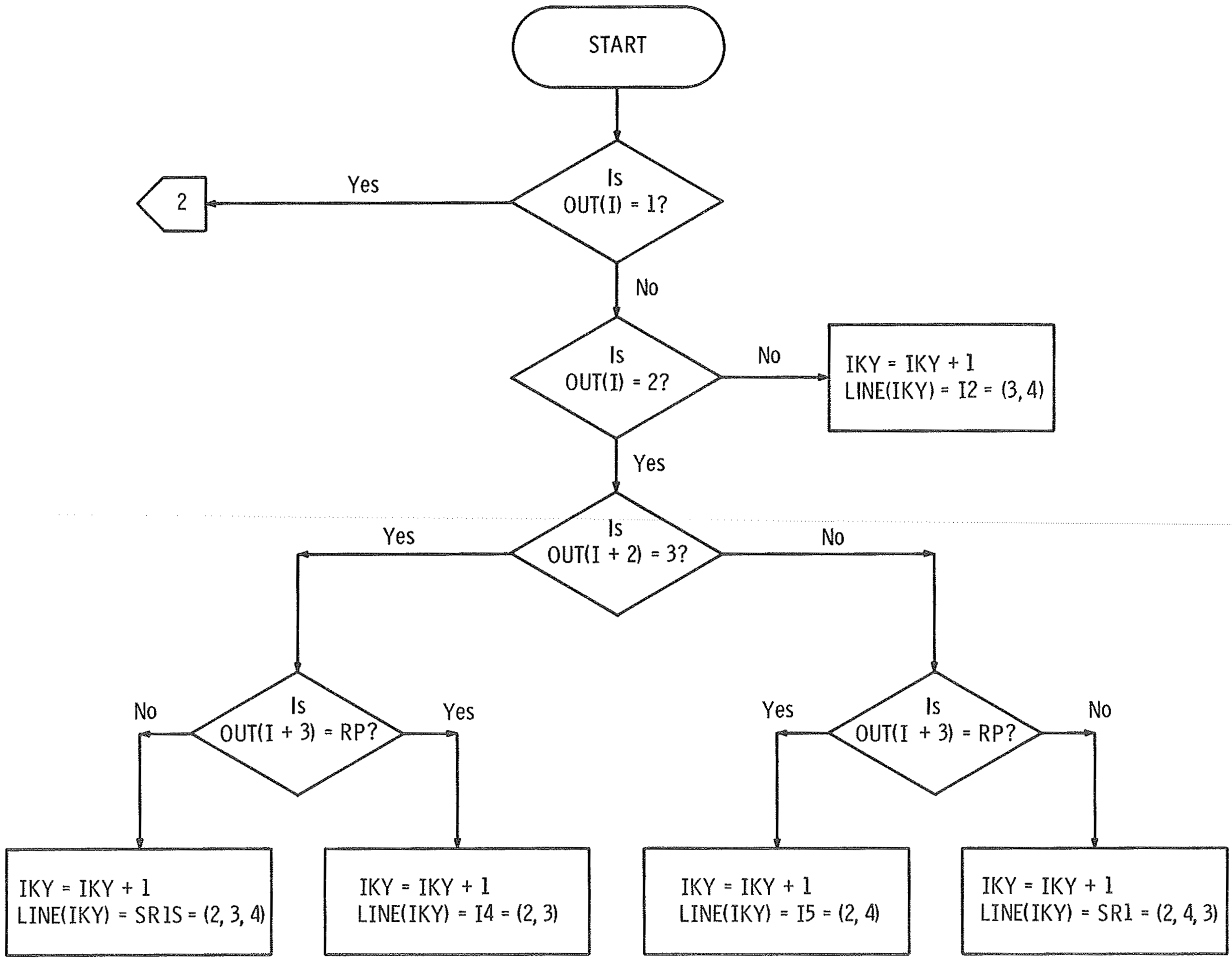
```
$IBFTC READ.
C      THIS SUBROUTINE CAN ACCOMODATE  BOTH   S3 AND S4
C
      SUBROUTINE READ(INPUT,LINE,LTAG)
      DIMENSION INPUT(15),LINE(576),IGROUP(6)
      INTEGER SR1,SR2,SR3,SR4,SR1S,SR2S,SR3S,SR4S,R1,R2,R3,R1S,R2S,R3S,
     .R1C,R2C,R3C
      INTEGER ONE, TWO, THREE, LP, RP
      DATA IDNT,LP,RP/1HE,1H(,1H)/
      DATA(IGROUP(I),I=1,3)/1H1,1H2,1H3/
      COMMON/IKYMAX/IKY,MAX,NCHK
      COMMON/NUMOPS/JO,KO,NOUT(24,24)
      COMMON/LITRAL/II1,II2,I3,I4,I5,I6,SR1,SR2,SR3,SR4,SR1S,SR2S,SR3S,
     .SR4S,R1,R2,R3,R1S,R2S,R3S,R1C,R2C,R3C
      ONE=IGROUP(1)
      TWO=IGROUP(2)
      THREE=IGROUP(3)
      IF(LTAG .EQ. 1) GO TO 1
      IF(NOUT(JO,KO) .NE. 0) GO TO 1
      IKY=IKY+1
      LINE(IKY)=IDNT
      GO TO 1900
    1 DO 1800 I=1,NCHK
      IF(INPUT(I)   .EQ. LP) GO TO 1800
      IF(INPUT(I)   .EQ. ONE) GO TO 1100
      IF(INPUT(I)   .EQ. TWO) GO TO 1010
      IF(INPUT(I)   .NE. THREE) GO TO 1800
      IKY=IKY+1
      LINE(IKY)=II2
      GO TO 1900
 1010 IF(INPUT(I+2)   .EQ. THREE) GO TO 1020
      IF(INPUT(I+3)   .EQ. RP) GO TO 1011
      IKY=IKY+1
      LINE(IKY)=SR1
      GO TO 1900
 1011 IKY=IKY+1
      LINE(IKY)=I5
      GO TO 1900
 1020 IF(INPUT(I+3)   .EQ. RP) GO TO 1021
      IKY=IKY+1
      LINE(IKY)=SR1S
      GO TO 1900
 1021 IKY=IKY+1
      LINE(IKY)=I4
      GO TO 1900
 1100 IF(INPUT(I+2)   .EQ. TWO) GO TO 1200
      IF(INPUT(I+2)   .EQ. THREE) GO TO 1130
      IF(INPUT(I+3)   .NE. RP) GO TO 1110
      IF(INPUT(I+4)   .EQ. LP) GO TO 1101
      IKY=IKY+1
      LINE(IKY)=I3
      GO TO 1900
 1101 IKY=IKY+1
      LINE(IKY) = R3S
      GO TO 1900
 1110 IF(INPUT(I+4)   .EQ. TWO) GO TO 1120
      IF(INPUT(I+5)   .EQ. RP) GO TO 1111
```

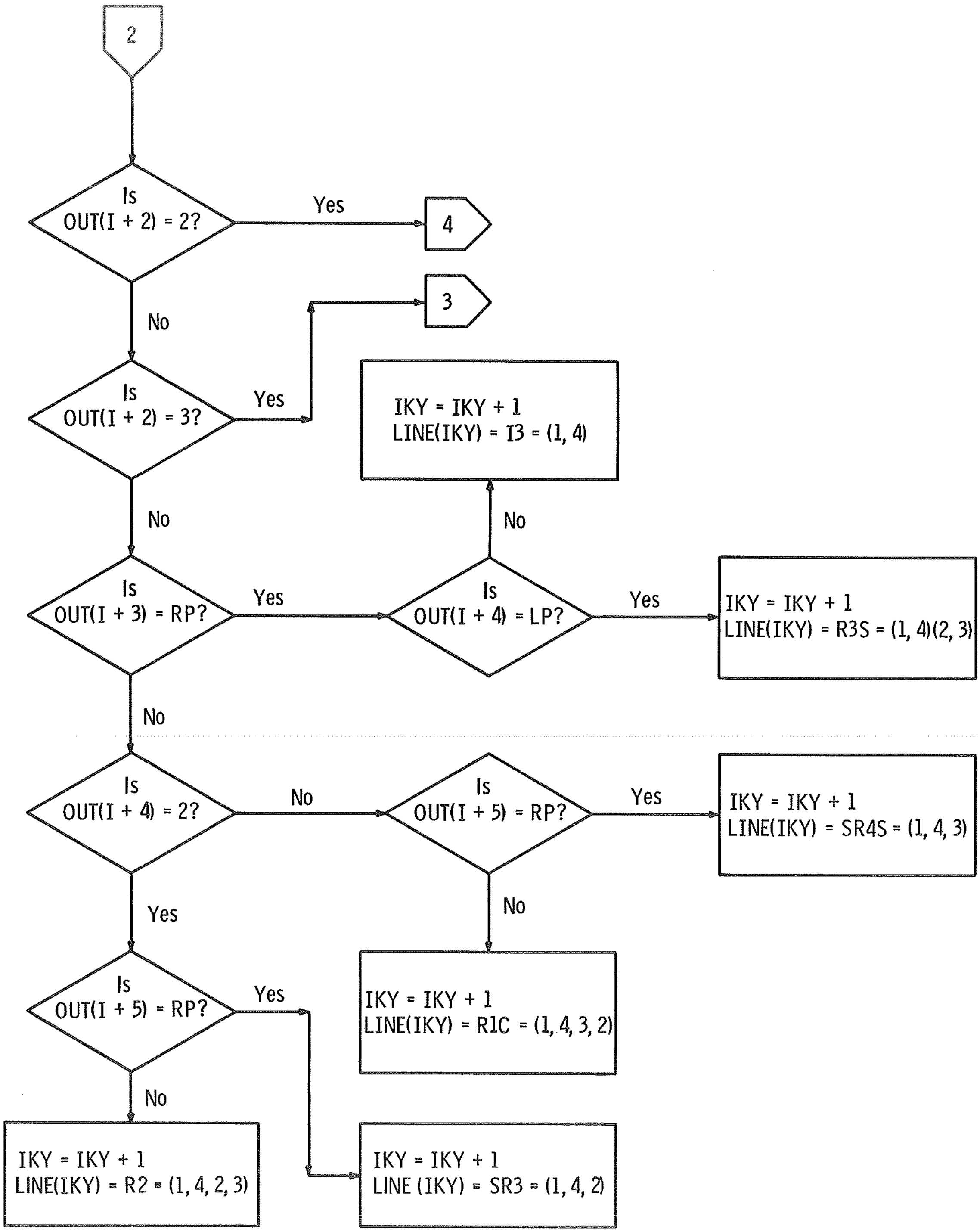
```
      IKY=IKY+1
      LINE(IKY)=R1C
      GO TC 19C0
 1211 IKY=IKY+1
      LINE(IKY) = SR4S
      GO TO 19C0
 1120 IF(INPUT(I+5)   .EQ. RP) GO TO 1121
      IKY=IKY+1
      LINE(IKY)=R2
      GO TO 19C0
 1121 IKY=IKY+1
      LINE(IKY)=SR3
      GO TC 19C0
 1130 IF(INPUT(I+3)   .NE. RP) GO TO 1140
      IF(INPUT(I+4)   .EQ. LP) GO TO 1131
      IKY=IKY+1
      LINE(IKY)=I6
      GO TC 19C0
 1131 IKY=IKY+1
      LINE(IKY)=R1S
      GO TC 19C0
 1140 IF(INPUT(I+4)   .EQ. TWO) GO TO 1150
      IF(INPUT(I+5)   .EQ. RP) GO TO 1141
      IKY=IKY+1
      LINE(IKY)=R3
      GO TC 19C0
 1141 IKY=IKY+1
      LINE(IKY)=SR4
      GO TC 19C0
 1150 IF(INPUT(I+5)   .EQ. RP) GO TO 1151
      IKY=IKY+1
      LINE(IKY)=R2C
      GO TC 19C0
 1151 IKY=IKY+1
      LINE(IKY)= SR2S
      GO TC 19C0
 1200 IF(INPUT(I+3)   .NE. RP) GO TO 1210
      IF(INPUT(I+4)   .EQ. LP) GO TO 1201
      IKY=IKY+1
      LINE(IKY) = I1
      LINE(IKY)=II1
      GO TC 19C0
 1201 IKY=IKY+1
      LINE(IKY)=R2S
      GO TC 19C0
 1210 IF(INPUT(I+4)   .EQ. THREE) GO TO 1220
      IF(INPUT(I+5)   .EQ. RP) GO TO 1211
      IKY=IKY+1
      LINE(IKY)=R3C
      GO TC 19C0
 1211 IKY=IKY+1
      LINE(IKY)=SR3S
      GO TC 19C0
 1220 IF(INPUT(I+5)   .EQ. RP) GO TO 1221
      IKY=IKY+1
      LINE(IKY)=R1
      GO TC 19C0
 1221 IKY=IKY+1
      LINE(IKY)= SR2
```

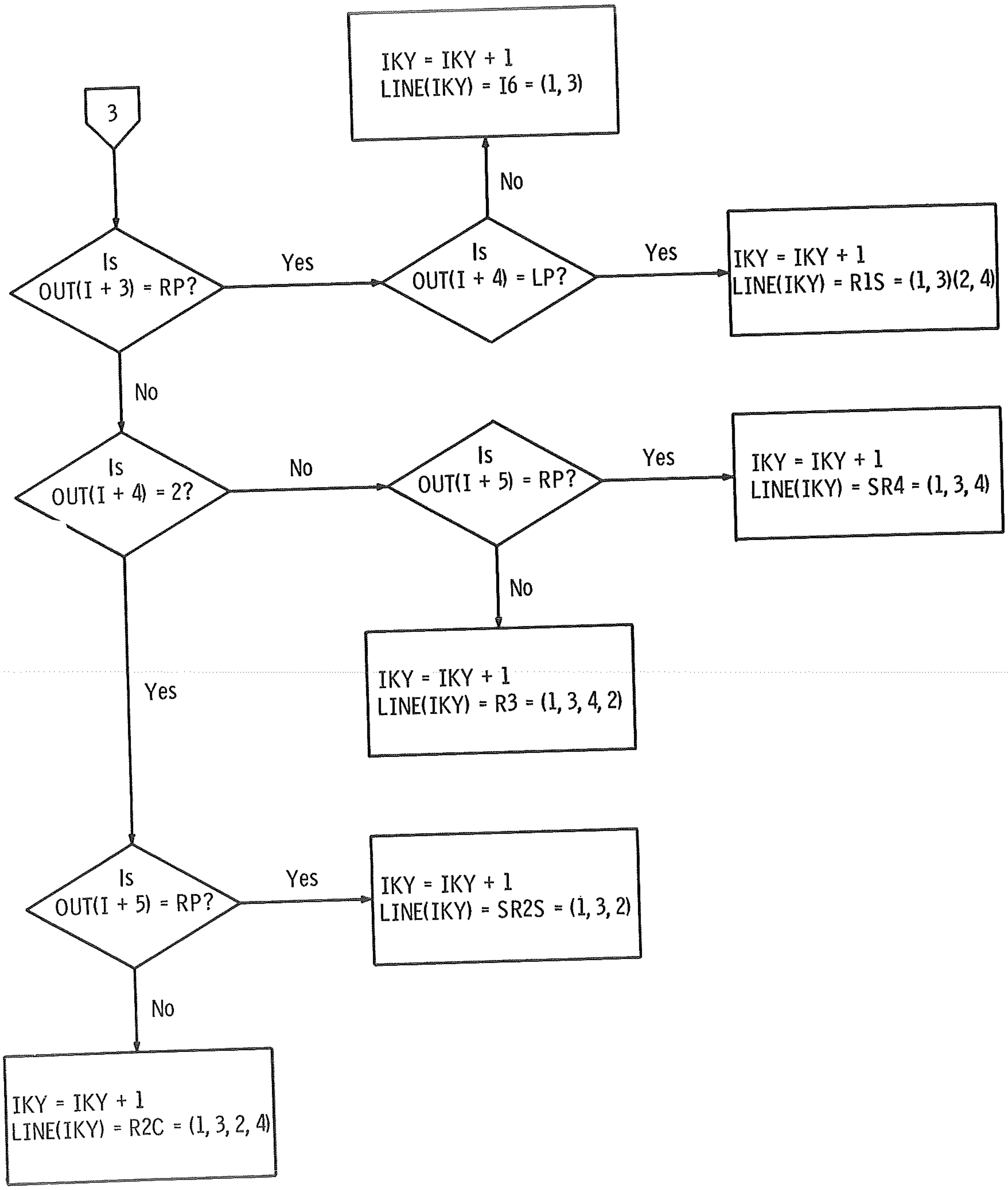
```
      GO TO 1900
 1800 CONTINUE
 1900 RETURN
      END
```

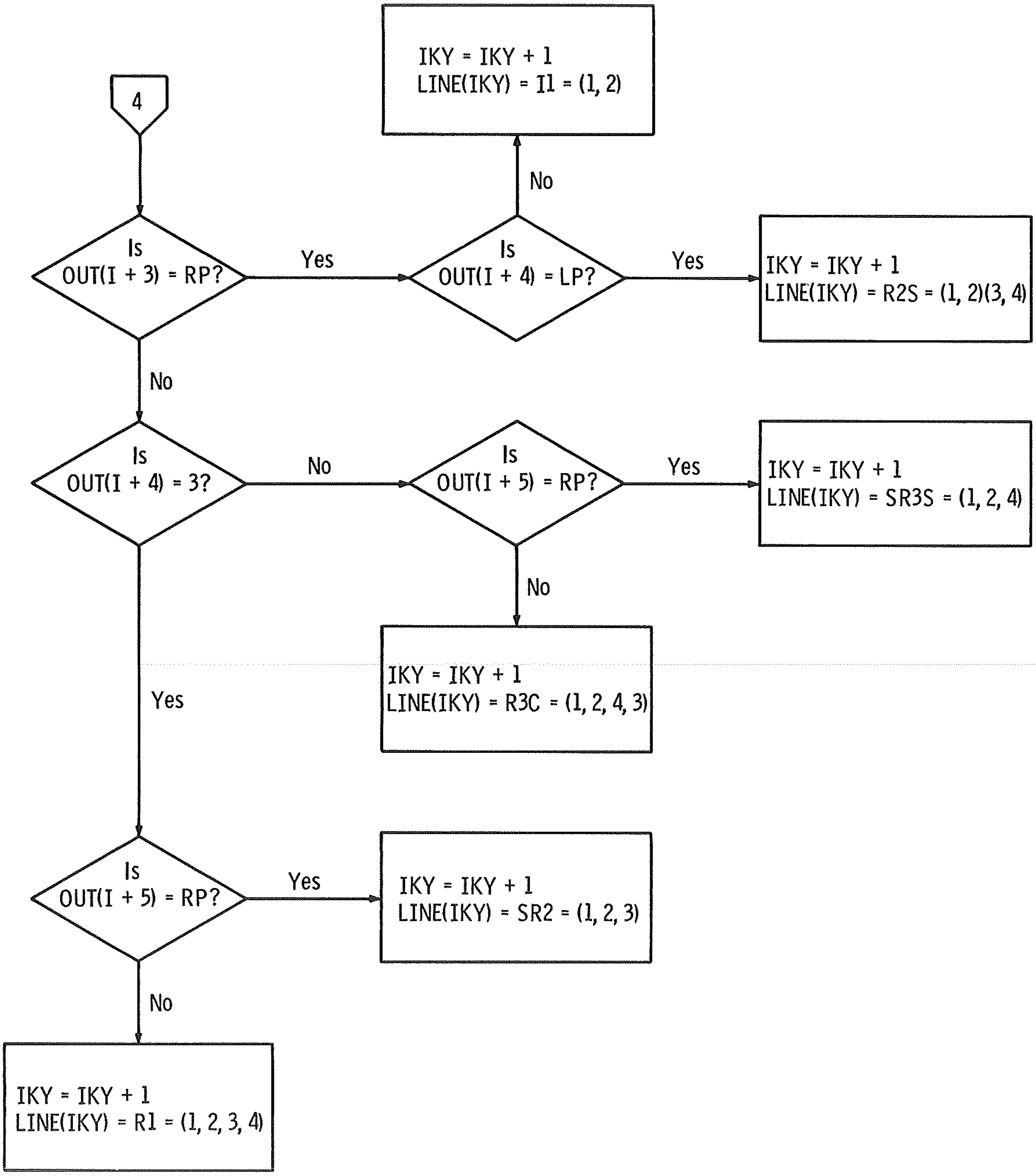
APPENDIX C

FLOW CHART OF SUBROUTINE READ




2
Is
OUT(I + 2) = 2?
Yes
4
No
3
Is
OUT(I + 2) = 3?
Yes
IKY = IKY + 1
LINE(IKY) = I3 = (1, 4)
No
No
Is
OUT(I + 3) = RP?
Yes
Is
OUT(I + 4) = LP?
Yes
IKY = IKY + 1
LINE(IKY) = R3S = (1, 4)(2, 3)
No
Is
OUT(I + 4) = 2?
No
Is
OUT(I + 5) = RP?
Yes
IKY = IKY + 1
LINE(IKY) = SR4S = (1, 4, 3)
Yes
No
Is
OUT(I + 5) = RP?
Yes
IKY = IKY + 1
LINE(IKY) = R1C = (1, 4, 3, 2)
No
IKY = IKY + 1
LINE(IKY) = R2 = (1, 4, 2, 3)
IKY = IKY + 1
LINE (IKY) = SR3 = (1, 4, 2)


3
Is
OUT(I + 3) = RP?
Yes
No
IKY = IKY + 1
LINE(IKY) = I6 = (1, 3)
Is
OUT(I + 4) = LP?
Yes
No
IKY = IKY + 1
LINE(IKY) = R1S = (1, 3)(2, 4)
Is
OUT(I + 4) = 2?
No
Yes
Is
OUT(I + 5) = RP?
Yes
No
IKY = IKY + 1
LINE(IKY) = SR4 = (1, 3, 4)
IKY = IKY + 1
LINE(IKY) = R3 = (1, 3, 4, 2)
Is
OUT(I + 5) = RP?
Yes
No
IKY = IKY + 1
LINE(IKY) = SR2S = (1, 3, 2)
IKY = IKY + 1
LINE(IKY) = R2C = (1, 3, 2, 4)


4
Is
OUT(I + 3) = RP?
Yes
No
Is
OUT(I + 4) = LP?
No
IKY = IKY + 1
LINE(IKY) = I1 = (1, 2)
Yes
IKY = IKY + 1
LINE(IKY) = R2S = (1, 2)(3, 4)
Is
OUT(I + 4) = 3?
No
Is
OUT(I + 5) = RP?
Yes
IKY = IKY + 1
LINE(IKY) = SR3S = (1, 2, 4)
No
IKY = IKY + 1
LINE(IKY) = R3C = (1, 2, 4, 3)
Yes
Is
OUT(I + 5) = RP?
Yes
IKY = IKY + 1
LINE(IKY) = SR2 = (1, 2, 3)
No
IKY = IKY + 1
LINE(IKY) = R1 = (1, 2, 3, 4)

REFERENCES

1. Lomont, John S.: Applications of Finite Groups. Academic Press, 1959.

2. Cannon, John J.: Computers in Group Theory: A Survey. Comm. ACM, vol. 12, no. 1, Jan. 1969, pp. 3-12.

3. Allen, Gabriel; Evans, David D.; and Swigert, Paul: FORTRAN Program for Machine Computation of Group Tables of Finite Groups. NAST TN D-5402, 1969.

4. Evans, David D.; and Allen, Gabriel: FORTRAN IV Program for Computation of Group Tables of Finite Groups - Program for Second Generation Machines. NASA TM X-2015, 1970.

5. Allen, Gabriel: An Efficient Method for Computation of Character Tables of Finite Groups. NASA TN D-4763, 1968. Note that there is an error on p. 27. In the i_4 row, the entries in the r_1^2 and r_2^2 columns should be interchanged. That is, group operation $i_4 \times r_1^2$ should result in i_2, whereas $i_4 \times r_2^2$ should be i_1.

NASA-Langley, 1971 —— 19 E-5927

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546