VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS

Michael Abbamonte
Selby H. Evans

November 1971
AMCMS Code 5011.11.85000

HUMAN ENGINEERING LABORATORIES

ABERDEEN RESEARCH & DEVELOPMENT CENTER
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.
VARGUS PATTERN SYNTHESIS TECHNIQUES AND THEIR APPLICATIONS

Michael Abbamonte
Selby H. Evans

November 1971

APPROVED:

JOHN D. WEISZ
Director
Human Engineering Laboratories

HUMAN ENGINEERING LABORATORIES
U. S. Army Aberdeen Research & Development Center
Aberdeen Proving Ground, Maryland

Approved for public release; distribution unlimited.
FOREWORD

This research was supported by the Department of Defense, Project THEMIS Contract (DAAD05-68-C-0176), under the Department of the Army, to the Institute for the Study of Cognitive Systems through the TCU Research Foundation. Further reproduction is authorized to satisfy needs of the U. S. Government.

The development of the VARGUS 7 program was supported by NIH Grant Number 11730-01.

The development of the VARGUS 9 program was supported in part by Grant Number MH-12223-01 from National Institute of Health and in part by THEMIS Contract DAAD05-68-C-0176.

The development of the VARGUS 10 program was supported in part by Grant Number MH-12223-01 from National Institute of Health and in part by THEMIS Contract DAAD05-68-C-0176.
ABSTRACT

This document describes a collection of computer programs developed for use in research on human pattern perception. The overall orientation which guided the development of the VARGUS (Variable Generator of Unfamiliar Stimuli) pattern-generation programs and the historical backgrounds of each category (VARGUS 7, 9 and 10) are related in the first section. The second section provides documentation, sample output and summary for each program and subroutine.
CONTENTS

FOREWORD .......................................................... ii
ABSTRACT ........................................................... iii

SECTION I. THREE PATTERN-GENERATION SYSTEMS: THEIR
CHARACTERISTICS AND APPLICATIONS .............................. 1

VARGUS 7 ............................................................... 2
VARGUS 9 ............................................................... 5
VARGUS 10 .............................................................. 9

REFERENCES ........................................................... 13

SECTION II. INTRODUCTION TO VARGUS PATTERN
GENERATION PROGRAMS ............................................... 17

APPENDIX ............................................................. 19

FIGURES
1. Six Examples of Vargus 7 Patterns Plotted as Histograms and as Graphs .... 4
2. Examples of Vargus 9 Patterns in Histograms and Graphic Format .......... 7
3. Examples of Polygons Produced by a Modification of the Vargus 9
System Developed by Aiken, L. S. & Brown, D. R. ............................. 8
4. Example of a Drawing Prepared by an Artist Under Instructions to
Produce Familiar Figures Under Constraints Described in the Text ........... 10
5. Examples of Vargus 10 Patterns ................................... 11

TABLES
1. Vargus 7 Control Table: Selected Redundancy Values and
Corresponding Values of $\frac{H}{H}$ and $p_{\text{max}}$ ......................... 2
2. Vargus 9 Control Table: Values of $p_{\text{max}}$ to Achieve Selected Levels
of Constraint Redundancy ............................................... 6

v
SECTION I

THREE PATTERN-GENERATION SYSTEMS:
THEIR CHARACTERISTICS AND APPLICATIONS

This collection of programs has been developed for use in research on pattern perception. The overall orientation which guided the development of these particular programs is schema theory as described by Evans (22). A pattern, in the context of schema theory, is a set of common characteristics or attributes associated with things belonging to the same pattern class. Individual instances of the class would not be identical nor would a particular instance necessarily exhibit all of the characteristics of the pattern.

A good example of such a pattern is a person's handwriting. Each person has a number of distinctive features associated with his handwriting. No feature is perfectly reliable in the sense that it is reproduced exactly in each handwriting sample. But taken together, the set of features is sufficient to permit a person's handwriting to be recognized or identified -- at least by experts.

The programs were intended to provide components of a methodology for studying human pattern perception. They are designed to offer the researcher options in choosing levels of familiarity and in controlling relevant stimulus parameters. To a substantial degree, these two options are mutually exclusive, because familiar stimuli are presumably encoded by subjects in terms of previously learned characteristics. Thus a determination of the relevant parameters would have to be based on a knowledge of human encoding processes -- a knowledge which is, of course, not yet available. Unfamiliar patterns, on the other hand, can be designed in such a way that the subjects will find and respond to the parameters the experimenter has chosen to make relevant. (This assertion is supported in the studies discussed in later sections of this paper.) These generating systems are, therefore, useful in providing a range of choice with respect to these two properties.

The programs seem to have some fairly general utility for perception research, and this laboratory has received a number of requests either for programs or for copies of the stimuli. This document has been prepared to make the programs themselves more generally available. None of the programs are particularly difficult or complex. Most of them could be written rather quickly by any reasonably competent programmer. Nevertheless, experimenters may find it convenient to have the programs already available and debugged.

This set of programs may be subdivided into three general categories depending on the main pattern-generation program. These categories are: VARGUS 7, VARGUS 9 and VARGUS 10 (the term VARGUS was originally an acronym for Variable Generator of Unfamiliar Stimuli). In subsequent sections, each of the generating systems will be described in detail and some of its applications will be noted.
VARGUS 7

The VARGUS 7 system was designed to produce very unfamiliar patterns, but ones which allowed for substantial control and manipulation of relevant pattern parameters. The theory behind the system is given by Evans (24). The VARGUS 7 generates segments of a Markov process, and the theory of Markov processes is used to achieve a number of features which might be desirable in the study of pattern perception. It allows the sampling of unfamiliar stimuli from a defined population and permits the independent control of uncertainty (in bits per stimulus) and of redundancy. In the long run, all pattern elements occur with equal frequency in every pattern position (1st element, 2nd element, etc.). Moreover, one schema can be substituted for another without altering these characteristics.

The schema is expressed in the form of favored transitional probabilities from element to element; taken together, the set of favored elements constitutes a Most Probable Sequence (MPS). Different schemata are produced by choosing different permutations of the elements to constitute the MPS. The common characteristics of VARGUS 7 patterns are thus local features (pairs or triples of elements), and these are the only features that can be used to identify a particular pattern.

In the basic generating program, the elements of the Markov process are the integers from 1 to 7. The resulting instances are strings of integers; final stimuli are produced by mapping the integers into other kinds of elements, as described below.

Table 1 gives parameter values to achieve selected levels of redundancy in generating VARGUS 7. The value $p_{max}$ is used as the probability of the favored transition for each element; each of the other transitions is assigned a probability $(1 - p_{max})/6$. Under these conditions, the redundancy and uncertainty of the Markov process are as given in Table 1.

**TABLE 1**

VARGUS 7 Control Table: Selected Redundancy Values and Corresponding Values of $H$ and $p_{max}$

<table>
<thead>
<tr>
<th>Redundancy (Percent)</th>
<th>$H$ (Bits per Column)</th>
<th>$p_{max}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2.81</td>
<td>.1428</td>
</tr>
<tr>
<td>20</td>
<td>2.24</td>
<td>.52</td>
</tr>
<tr>
<td>30</td>
<td>1.97</td>
<td>.61</td>
</tr>
<tr>
<td>40</td>
<td>1.69</td>
<td>.69</td>
</tr>
<tr>
<td>50</td>
<td>1.42</td>
<td>.76</td>
</tr>
<tr>
<td>60</td>
<td>1.10</td>
<td>.83</td>
</tr>
<tr>
<td>70</td>
<td>.84</td>
<td>.88</td>
</tr>
<tr>
<td>80</td>
<td>.55</td>
<td>.93</td>
</tr>
</tbody>
</table>
VARGUS 7 stimuli have been used most commonly by mapping the strings of numbers into column heights in histograms or altitudes in graphs (Fig. 1). In this latter form, the stimuli have some resemblance to time series data and their common characteristics are like transients occurring at random times.

These stimuli have been used in a substantial number of studies calling for subjects to distinguish between stimuli representing different schemata. It has been well established that subjects can accomplish this task (26, 15, 25, 19, 7).

Rosser (37) presented stimuli similar to those produced by the VARGUS 7 in an auditory format using the elements of the Markov processes to determine particular tone; she also found discrimination between classes of patterns, as did Copeland (13) with similar auditory patterns generated by the VARGUS 7 system. More recently Pollack (34) translated similar Markov processes into pulse trains, converted them to sound, and again demonstrated that subjects could spontaneously discriminate between classes of patterns. Similar effects were demonstrated by Holler and Evans (31), using a language-like mode of presentation in which the elements of the Markov process were mapped into syllables.

The stimuli generated by the VARGUS 7 system have also been used in modeling efforts designed to emulate the behavior described above (21, 4, 8).

In addition to the basic demonstration of schematic concept formation, the VARGUS 7 stimuli have been used in a number of studies investigating the effect of constraint redundancy (23) and the effect of knowledge of results on schematic concept formation (10, 18, 20, 11, 39, 5). These studies have tended to show that constraint redundancy is an important variable in this context, and that schematic concept formation is accomplished reasonably well at 70 percent redundancy, marginally at 50 percent redundancy, and probably not at all at 40 percent redundancy. These figures, of course, apply to the kinds of stimuli used in the experiments-standard VARGUS 7 stimuli containing 12 to 24 elements and represented as histograms. There is no reason to expect a redundancy measure to have enough generality to permit its application to other kinds of patterns.

The effects of knowledge of results appear to be a bit more complex, but there is some suggestion (3, 41) that knowledge of results is ineffective and may actually interfere with performance when the patterns are sufficiently redundant (e.g., 70%). At lower redundancies, knowledge of results appears to facilitate performance.

Schema theory proposes that subjects remember an individual stimulus by encoding it in terms of its deviations from the schema. The VARGUS 7 stimuli have been used in studies of this question (16, 17, 32); there appears to be some evidence in support of the schema-plus-correction hypothesis when reproduction tasks are used. On the other hand, a study by Brown and Rebbin (10) tends to suggest that in pattern-classification tasks, at least, encoding of the schema plays very little role in the classification of the VARGUS 7 stimuli. Since encoding the schema is presumably a prerequisite to memory storage in the form of schema plus correction, further research on this point seems to be needed.

A set of VARGUS 7 patterns in numerical form has been prepared and made available (2); for many research purposes this standard set of VARGUS 7 stimuli would be suitable. That monograph also described a measure which allows each stimulus to be represented in terms of a single number indicating how closely it conforms to the prototype. This measure is computed by the subroutine ADHER, listed along with VARGUS 7.
Fig. 1. SIX EXAMPLES OF VARGUS 7 PATTERNS PLOTTED AS HISTOGRAMS AND AS GRAPHS
(The histoform plots were produced by an IBM 1401 printer using the suppress-space feature to overprint the characters zero, capital O and asterisk. Each row represents a different schema.)
VARGUS 9

The VARGUS 9 stimuli are generated in accordance with a method described by Evans and Mueller (27). As with the VARGUS 7, the stimuli are randomly sampled from a defined population, with information and redundancy subject to control and manipulation. The following additional objectives were also achieved:

1. The population is made up of variants which consist of independent and measurable deviations from a defined prototype.

2. It is reasonable to treat the deviations as interval scale deviations from a mean, so that interval scale measures can be applied.

In the VARGUS 9 system a prototype is a string of integers ranging in value from 3 to 7. Any such string of no more than 32 elements will serve, although one would normally choose the string so as to avoid any systematic ordering of the integers. Variants are produced by adding to the prototype integer a deviation term, which is an integer ranging from -2 to +2. The probability distribution for this deviation is unimodally and symmetrically distributed about zero.

The procedure for determining the distribution of the deviation, as suggested by Evans and Mueller (27), is to choose \( p(0) = \frac{1}{6} \) and then set the other values as follows:

\[
p(1) = \frac{(1 - p_{\text{max}})}{3}, \quad p(2) = \frac{(1 - p_{\text{max}})}{6}.
\]

Under these rules, the required values of \( p_{\text{max}} \) for selected redundancy levels are as given in Table 2. In general, the redundancy levels for exhibiting schematic concept formation are similar to those for VARGUS 7; 50 percent redundancy is marginal, while 70 percent gives relatively good results. As with the VARGUS 7, these comments apply only to the kinds of stimuli used in the studies reported below.

VARGUS 9 stimuli have been studied principally in displays mapping their numbers into column heights of histograms or into altitudes on a graph (Fig. 2). In this form, the stimuli might be likened to personality profiles or to other graphic displays in which the variability from column to column is relatively independent. In contrast to the VARGUS 7, the common characteristics of the VARGUS 9 stimuli lie in global features and in the overall shape. While local features are to some extent usable, they are less reliable than in comparable VARGUS 7 stimuli.

These stimuli have also been used in studies of schematic concept formation (35, 29, 30). Again, the research has shown schematic concept formation to occur and there is some possibility that it is easier with VARGUS 9 stimuli than with VARGUS 7 stimuli (40). These stimuli have also been used in polar-coordinate form (33, 1) in which the numbers constituting the patterns are plotted as length of radii. In this form they constitute a closed polygon, and research with them has provided a link between schematic concept formation studies and the Brown and Owen psychophysics of form (12). Aiken and Brown (1) have extended this research with a modified version of the VARGUS 9 pattern-generating system (Fig. 3). One of the major results of this research has been the demonstration of subjects' sensitivity to the statistical distributions of features within the particular set of patterns on which judgments are being made. Similar effects have been shown by Dansereau, Fenker and Evans (14), who demonstrated that pattern storage in memory is substantially influenced by the clusters of patterns constituting schema families.
The perceived similarity of VARGUS 9 stimuli, when they are drawn from the same cluster, or schema family, appears to be determined largely by the Euclidean distance between the pair of stimuli (36); a correlation of .88 was found between judged similarity and interpoint distance. (This quantity is calculated by the subroutine DVAR, listed with VARGUS 9.) The strength of this relationship supports the assertion that interval scale measures are appropriate with VARGUS 9 stimuli and suggests that these stimuli might have some general usefulness in cases where it is desirable to control the subjective similarity of patterns.

### TABLE 2

VARGUS 9 Control Table: Values of $p_{\text{max}}$ to Achieve Selected Levels of Constraint Redundancy

<table>
<thead>
<tr>
<th>Redundancy</th>
<th>$p_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>30%</td>
<td>.31</td>
</tr>
<tr>
<td>40%</td>
<td>.53</td>
</tr>
<tr>
<td>50%</td>
<td>.66</td>
</tr>
<tr>
<td>60%</td>
<td>.77</td>
</tr>
<tr>
<td>70%</td>
<td>.84</td>
</tr>
</tbody>
</table>
Fig. 2. EXAMPLES OF VARGUS 9 PATTERNS IN HISTOGRAMS AND GRAPHIC FORMAT
(Each column represents a different schema. The histograms and the
topmost graphs are prototypes. Instances below these represent
increasing deviations from the prototypes.)
Fig. 3. EXAMPLES OF POLYGONS PRODUCED BY A MODIFICATION OF THE VARGUS 9 SYSTEM DEVELOPED BY AIKEN, L. S. & BROWN, D. R. (1)
The previous generating programs produced stimuli that were intended to be relatively unfamiliar. The VARGUS 10 system was designed to provide stimuli which are of intermediate familiarity. In particular, the VARGUS 10 patterns are intended to incorporate relatively familiar components into overall configurations which are relatively unfamiliar.

The method for obtaining familiar configurations was based on a method devised by Shannon and Weaver (38). In studying the transitional probabilities of English grammar, Shannon used humans as repositories of the statistical characteristics of grammar. In the same way, one might expect that humans are repositories of the statistical characteristics of the visual environment and that they use these characteristics when they draw figures. In accord with that logic, a commercial artist was engaged to draw a number of figures in such a way as to allow them to be decomposed into interchangeable components.

The original drawings were made in a 3 x 3 matrix (Fig. 4); the eight outer cells of the matrix had marks on the midpoints of the lines separating them from their adjoining cells. The artist was instructed to draw each figure by starting at one of the midpoints and drawing any line he pleased through the cell, so long as the line did not touch itself and did not cross the border of the cell except at the other midpoint, where the line could exit to the next cell. He was then to continue in the same fashion through the remaining cells until he returned to the starting point. He was encouraged to produce any familiar figures he could within those constraints.

The resulting drawings were then decomposed by taking the configuration in each cell as a separate unit. The matrix had four "corner" cells and four "side" cells. Since all corner cells, if properly rotated, had the same entry and exit points for the line, any corner cell could be used in any corner to construct a figure. The same condition applied to the side cells.

After the components were dissociated from their source figures, they were reviewed to eliminate components which were near duplicates of each other and to eliminate components which contained too much detail to be adequately represented by black dots in a 16 x 16 matrix. The resulting set—40 corner components and 34 side components—constituted the basic set out of which patterns could be generated.

The VARGUS 10 program selects components at random from these sets to fill each of the four corner cells and four side cells of a figure being constructed. The components are rotated to conform to the orientation of the assigned cells, and each component is then placed to comprise a 16 x 16 sub-matrix of a 48 x 48 matrix which contains the whole pattern (Fig. 5).

A number of associated programs have been prepared to manipulate the VARGUS 10 stimuli and to provide a measure—probably a crude one—of similarity. These programs are also provided in the VARGUS 10 section.

The VARGUS 10 generating system has not been the subject of as many published reports as have the VARGUS 7 and VARGUS 9. Stimuli generated with this system were used in conjunction with the elements of a model for human preprocessing by Zinser (42, 43), and they have been used in a study of feature selection by Hastings and Evans (28). This latter study established a methodology for obtaining judgments from people regarding the features they would use for discriminating patterns; the results indicated a preference for corner configurations and for projections.
Fig. 4. EXAMPLE OF A DRAWING PREPARED BY AN ARTIST UNDER INSTRUCTIONS TO PRODUCE FAMILIAR FIGURES UNDER CONSTRAINTS DESCRIBED IN THE TEXT
(The drawing was termed "Old Witch.")
Fig. 5. Examples of VARGUS 10 Patterns
A number of other studies of feature selection are possible with the VARGUS 10 stimuli; these studies have not been executed under the present research project only because other studies were of higher priority for the purposes of the project. For example, no study has been carried out to assess the adequacy of the similarity measure computed by CONG; doubtless, such a study would be of substantial interest.

One might assess the effectiveness of the overall similarity measure in predicting subjective similarity. One might also measure the similarity of the individual components and use these values in a regression equation to compare their predictive power with the predictive power of the overall similarity measure. (The difference between these two prediction methods is that the first assumes that the whole is the sum of its parts, while the second assumes that the whole is the best weighted linear combination of its parts.) One could also obtain subjective judgments of the similarities of the parts to determine the relative adequacy of these measures in predicting overall similarity as compared with the objective similarity measures. Such studies would shed additional light on the processes of feature selection and feature utilization.
REFERENCES


INTRODUCTION TO VARGUS PATTERN GENERATION PROGRAMS

The purpose of this introduction to the documentation is to familiarize the reader with the overall content of the programs that follow and to discuss certain programming considerations not relevant to the description of the programs and thus not discussed in the documentation. The following pages contain the documentation for 21 programs and subroutines that have been used for the computer generation, measurement, manipulation and construction of the VARGUS 7, 9 and 10 patterns. These programs have been used on many occasions in the past, and the program listings in the documentation are exactly those programs which produced the sample outputs.

All programs are written in IBM 1800 Basic FORTRAN IV except program PAT which is written in AUTOCODER for the IBM 1401. In principle, the FORTRAN programs should be translated readily by FORTRAN compilers, regardless of the particular machine and implementation, especially since a USA Standard Basic FORTRAN IV is an accepted standard for the language. In fact, implementations of compilers and semantics of program statements frequently are different enough to prevent direct communication of programs between machines. We will now discuss some of the more common incompatibilities which the user may experience in attempting to utilize IBM 1800 Basic FORTRAN IV on other machines and briefly indicate how to correct them when they occur.

Probably the primary source of incompatibility a user will experience with these programs is the mixed mode arithmetic expressions. Standard FORTRAN does not permit mixed-mode expressions. The semantics of mixed expressions contained in these programs are as follows:

a. All terms in a mixed expression involving only floating-point variables are evaluated with floating-point arithmetic.

b. All terms in a mixed expression involving only integer-mode variables are evaluated with integer arithmetic.

c. When a mixed expression contains terms that involve both integer and floating-point variables, the integer quantities are converted to floating point before performing the floating-point arithmetic. Consider the mixed expression $A \times B + I/J - C/(I + J)$. It is evaluated as follows:

a. $A \times B$ is evaluated in floating point, and the product is a floating-point number.

b. $I/J$ is an integer division so that any fractional remainder after division is lost, and the integer part of the quotient is saved as an integer-mode result.

c. In $C/(I + J)$, $(I + J)$ is evaluated with integer arithmetic, but before the division occurs the sum is converted to floating point, and the division is performed by floating-point arithmetic and yields a floating-point quotient.

d. The floating-point product of $A \times B$ is then added to the quotient of $I/J$ after the latter has been converted to floating point (the sum is a floating-point number), and finally,
e. The floating-point quotient of \( C/(1 + J) \) is then subtracted from the result of "d" to obtain a floating-point difference.

The user should note that an equivalent evaluation of this expression is not achieved by simply changing it to \( A \times B + \frac{FLOAT(I)}{FLOAT(J)} - \frac{C}{FLOAT(I) + FLOAT(J)} \) since the middle quotient now may include a fractional part. When every term in an expression is either mixed or floating point (no integer terms) such a transformation will suffice. The numerical result of any mixed expression is always in floating-point mode.

Another source of incompatibility may result from the use of apostrophes to enclose literal data in Format statements. An equivalent representation may be achieved by using the H-format code.

Finally the DATA specification statement which initializes variables at compile time is not a USA Standard Basic FORTRAN IV feature. If this is a basis for an incompatibility, the initialization may be done with an input statement at execution time.

There are other ways in which IBM 1800 Basic FORTRAN IV differs from USA Standard Basic FORTRAN IV but they should not be the basis for incompatibilities. For example, 1800 FORTRAN IV variable names are limited to a length of five characters, whereas the more common length is six characters. Other incompatibilities of this innocuous sort are not discussed here.

For the readers' information, the logical unit numbers referenced in input-output statements in the program are associated with the following devices: logical unit 2 is the card reader-punch, logical unit 3 is the line printer, logical units 8 and 9 are tape drives. When no format number is associated with input-output from tape, all data transfer is in binary and involves no conversion.

One brief note should be made to users who may desire to convert these programs to FORTRAN II. In some programs, Type specification statements are used which alter the implicit mode of a variable, for example, INTEGER PATR permits PATR to be used as an integer mode variables. Since FORTRAN II does not implement Type statements, the desired mode may be obtained by adding an appropriate leading letter to the variable name, for example, KPATR. Also FORTRAN IV input-output statements which reference logical-unit numbers will have to be changed to reference Format statements only, and in output statements in particular the Format statement will have to be adjusted if no carriage control is needed. The apostrophes to define literal data will also have to be changed to an H-format code. Finally the DATA statement which initialize certain variables at compile time will have to be deleted and changed to initialization by means of an input statement.

An overview of the programs will illustrate their organization and interrelationships. In the Appendix, the term "Program Name" indicates the associated routine is a main program and the term "Subroutine Name" indicates the routine is a subroutine program. To the right of each routine's name is a brief statement of its purpose. When a routine's name is indented to the right relative to the routine name above it, the indented name is a subroutine called by the routine above it. It should be noted that there are subroutines below without an associated calling routine. Since the calling routines were relatively trivial (restricted largely to doing input and output) they were selectively excluded. Finally, sample executions are not included for each and every subroutine, since in many instances their function is minor and supporting a larger goal. Sample runs are included only for each composite of routines with a major objective.
APPENDIX
SUMMARY OF VARGUS PROGRAMS

Vargus 7
Program Name: VARGUS 7E  
Program Name: ADHER  
Program Name: PAT  

Pattern generation.  
Pattern measurement.  
Pattern construction.  

Vargus 9
Program Name: VARGUS 9  
Program Name: VARG 9  
Program Name: DVAR  

Pattern generation and measurement.  
Pattern generation and measurement.  
Pattern measurement and construction.  

Vargus 10
Program Name: VARGUS 10A  
Subroutine Name: JOIN  
Subroutine Name: ROTAT  
Subroutine Name: MODUL  
Subroutine Name: CONG  
Subroutine Name: BLUR  
Subroutine Name: RSCAL  
Subroutine Name: MIMAX  

Input submatrices and output pattern image.  
Select submatrices.  
Rotate submatrices.  
Perform modular arithmetic.  
Measure pattern similarity.  
Blurs the input pattern.  
Rescale numbers in matrix.  
Find largest and smallest value in a matrix.  

19
Subroutine Name: SOLID

Fill in figure defined by a perimeter of ones.

Subroutine Name: WIDEN

Widen the lines of a figure.

Subroutine Name: WIDER

Widen lines of a closed figure inwardly.

Subroutine Name: SOLID
Subroutine Name: WIDEN

General Purpose Programs

Subroutine Name: RANDY

Pseudo-random number generator.

Subroutine Name: PATOT

Output pattern matrix.
I. Identification

1. Program Name: VARGUS 7E
2. Category: Pattern Generation
3. Purpose: Generate strings of numbers containing first-order sequential (Markov) dependency.
4. Date: September 8, 1970
5. Programmer: Selby Evans

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Elements of the Markov process are the numbers 1 to 9.
   Reuse of the same START card for the random-number generator will produce the same number sequence.
4. Input Data:
   The first call to Subroutine RANDY results in the START card being read in. See RANDY documentation for details of its usage.
   LNTH: The number of digits constituting a single pattern.
   INTL: The identification number assigned to the first instance.
   INST: The number of pattern instances desired.
   LEAD: The first pattern element (digit from 1 to 9) in each pattern. If LEAD = 0, the first
pattern element is chosen at random for each pattern.

NCØL: Number of elements in the Markov process.
Maximum allowable in this program is NCØL = 9.

P: An NCØL x NCØL Markov matrix. Each data-card input to P is one row of P.

5. Output Data:
Final pattern is contained in array L and is output on the printer.

6. Subroutines Required:
RNDY: A pseudo-random number generator.

III. Mathematical Methods and References

IV. Listing of Source Program
Attached.

V. Sample Run
Attached.

VI. Remarks
None.
// JOB 11000 30 JUN 71 13.404 HRS
// * U-2050-003
// FOR ABBA 30 JUN 71 13.405 HRS
*IOCS (CARD,1443 PRINTER, PLOTTER, DISK, MAGNETIC TAPE, KEYBOARD, TYPEWR
*NONE PROCESS PROGRAM
*LIST ALL
*ONE WORD INTEGERS
C VARGUS 7-E
  DIMENSION P(9,9), L(60)
  CALL RANDY(R)
  WRITE(3,901)
  901 FORMAT(*11,T7,*SAMPLE VARGUS SEVEN PATTERNS*)
  WRITE(3,911)
  911 FORMAT(*00,T9,*ID*,T21,*PATTERNS*,/)
C C ENTER CONTROL CARDS AND MATRIX
  READ(2,4) LNTH, INTL, INST, LEAD, NCOL
  4 FORMAT(515)
  INST = INST + INTL - 1
  DO 10 I=1,NCOL
  10 READ(2,5) (P(I,J), J=1,NCOL)
  5 FORMAT(9F8.4)
C C SET UP POPULATION LOOP
  X=NCOL
  X=1.0/X
  DO 90 NDO = INTL, INST
  90 K=1,40
  L(K)=0
C C CHOOSE STARTING ELEMENT
  MARK = LEAD
  IF(LEAD) 30,30,50
  30 CALL RANDY(R)
  DO 40 MARK =1, NCOL
  40 R = R - X
  IF(R) 50,50,40
  CONTINUE
  MARK = NCOL
C C SET UP INSTANCE LOOP
  50 DO 80 IDO =1, LNTH
  80 L(IDO) = MARK
C C MOVE ONE STEP
  CALL RANDY(R)
  DO 60 K =1, NCOL
  60 R = R - P(MARK, K)
  IF(R) 70,60,60
  CONTINUE
  K = NCOL
  70 MARK = K
  80 CONTINUE
C PRINT RESULTING PATTERN AND ID ON PRINTER
WRITE(3,9) NDO, (L(I),I=1,LNTH)
9 FORMAT(4X,17,3X,12(1X,4I1))
90 CONTINUE
CALL EXIT
END

VARIABLE ALLOCATIONS

P(R )=00A0-0000  R(R )=00A2  X(R )=00A4  L(I )=00E1
INST(I )=00E4  LEAD(I )=00E5  NCOL(I )=00E6  I(I )=00E7
K(I )=00EA  MARK(I )=00EB  IDO(I )=00EC

STATEMENT ALLOCATIONS

901 =00F0 911 =0108 4 =0115 .5 =0118 9 =0118 10 =0149 20
60 =01DD 70 =01EA 80 =01EE 90 =0210

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS
RANDY FSUB FSUBX FDIV FLD FSTO FLOAT ISTOX LFAC MR
SUBSC FCHRI TYPEN HOLEB MAGT PRNTN EBPRT CARDN

REAL CONSTANTS
.100000E 01=00EE

INTEGER CONSTANTS
3=00F0 2=00F1 1=00F2 40=00F3 0=00F4

CORE REQUIREMENTS FOR ABBA
COMMON 0 INSKEI COMMON 0
VARIABLES 238 PROGRAM 302

END OF COMPILATION
<table>
<thead>
<tr>
<th>ID</th>
<th>PATTERNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1731 1725 3153 45</td>
</tr>
<tr>
<td>1002</td>
<td>7263 1727 2664 53</td>
</tr>
<tr>
<td>1003</td>
<td>4531 7726 7645 37</td>
</tr>
<tr>
<td>1004</td>
<td>5353 1726 4531 74</td>
</tr>
<tr>
<td>1005</td>
<td>3517 2645 3453 17</td>
</tr>
<tr>
<td>1006</td>
<td>7264 1717 2617 26</td>
</tr>
<tr>
<td>1007</td>
<td>7253 1726 4531 76</td>
</tr>
<tr>
<td>1008</td>
<td>2647 2645 1726 45</td>
</tr>
<tr>
<td>1009</td>
<td>1726 4574 5726 45</td>
</tr>
<tr>
<td>1010</td>
<td>5317 2641 7131 56</td>
</tr>
</tbody>
</table>
I. Identification

1. Program Name: ADHER
2. Category: Pattern Measurement.
3. Purpose: The program measures the degree to which a pattern generated by program VARGUS 7E adheres to the schema rule from which it was generated. See Section III below.
4. Date: May 17, 1971
5. Programmer: Mike Abbamonte

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: As written, the program will measure patterns generated from a seven-element Markov Process. Adjustment in the size of arrays MPS and MX will make the program more general.
4. Input Data:

NCOL: The length (number of digits in) an individual pattern.

LMPS: The length of the most probable sequence before cycling. Should be seven here.

NPAT: The number of patterns to be measured.

MPS: A vector containing the schema i.e., the most probable sequence.
IDPAT: Individual pattern identification number.
IN: The array containing the pattern.

5. Output Data:
IDPAT: Same as above.
IN: Same as above.
PSS: The proportion of schematic steps. See Section III for details.

6. Subroutines Required:
None.

III. Mathematical Method and References
The proportion of schematic steps (PSS) is the measure used to indicate the degree to which a pattern adheres to its generation rule (also called the MPS, i.e., most probable sequence). PSS is simply that fraction of the number of elements in a pattern which are followed by the most probable successor element.


IV. Listing of Source Program
Attached.

V. Sample Run
Attached.

VI. Remarks
None.
// JOB  30 JUN 71 13.438 HRS  
// FOR ABBA  30 JUN 71 13.438 HRS  
*NONE PROCESS PROGRAM  
*ONE WORD INTEGERS  
*IOCS (CARD, 1443 PRINTER, PLOTTER, DISK, MAGNETIC TAPE, KEYBOARD, TYPEWRITER)  
*LIST ALL  
C ADHER  
DIMENSION IN(24), MPS(7), MX(7,7)  
READ(2,2) ICOL, LMPS, NPAT  
2 FORMAT(315)  
FCOL=ICOL  
READ(2,3) (MPS(I), I=1,LMPS)  
3 FORMAT(2013)  
WRITE(3,88)  
88 FORMAT(11',T3','ID',T28,'VARGUS SEVEN PATTERNS',T60,'PROPORTION OF  
1SCHEMATIC STEPS','/')  
DO 500 I=1,NPAT  
READ(2,5) IDPAT, (IN(J), J=1,NCOL)  
5 FORMAT(15,5X,28(11,1X))  
DO 200 K1=1,LMPS  
DO 200 K2=1,LMPS  
200 MX(K1,K2)=0  
DO 100 L=1,LMPS  
IF(L-NCOL) 60,70,60  
60 IC=IN(L+1)  
GO TO 100  
70 IR=IN(L)  
100 MX(IR,IC)=MX(IR,IC)+1  
ICNT=0  
DO 150 L=1,LMPS  
IR=MPS(L)  
IF(L-LMPS) 120, 130s120  
130 IC=MPS(L)  
GO TO 150  
120 IC=MPS(L+1)  
150 ICNT=ICNT+MX(IR,IC)  
FCNT=ICNT  
PSS=FCNT/FCOL  
WRITE(3,30) IDPAT, (IN(L)*L=lsNCOL)* PSS  
30 FORMAT(15,10X,1413,12X,F8.4)  
500 CONTINUE  
CALL EXIT  
END  
  
VARIABLE ALLOCATIONS  
FCOL(R )=0000  
NCOL(I )=0056  
K1(I )=005C  
FCNT(R )=0002  
LMPS(I )=0057  
K2(I )=005D  
PSS(R )=0004  
NPAT(I )=0058  
L(I )=005E  
IN(I )=0010  
I(I )=0059  
IR(I )=005F  
  
STATEMENT ALLOCATIONS  
2 =0068  3 =0068  88 =006E  5 =0093  30 =009A  200 =00F0  60  
120 =0166  150 =016F  500 =01AB  

28
FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS
FDIV  FLD  FSTD  FLOAT  ISTOP  MRED  MWRT  WCMP  MIAIX  MICO
MAGT  PRNTN  EBPRT  CARDN

INTEGER CONSTANTS
2=0064  1=0065  3=0066  0=0067

CORE REQUIREMENTS FOR ABBA
COMMON  0  INSKEL  COMMON  0
VARIABLES  100  PROGRAM  338

END OF COMPILATION
<table>
<thead>
<tr>
<th>ID</th>
<th>VARGUS SEVEN PATTERNS</th>
<th>PROPORTION OF SCHEMATIC STEPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1001</td>
<td>1 7 3 1 1 7 2 5 3 1 5 3 4 5</td>
<td>0.5714</td>
</tr>
<tr>
<td>1002</td>
<td>7 2 6 3 1 7 2 7 2 6 6 4 5 3</td>
<td>0.7142</td>
</tr>
<tr>
<td>1003</td>
<td>4 5 3 1 7 7 2 6 7 6 4 5 3 7</td>
<td>0.6428</td>
</tr>
<tr>
<td>1004</td>
<td>5 3 5 3 1 7 2 6 4 5 3 1 7 4</td>
<td>0.7857</td>
</tr>
<tr>
<td>1005</td>
<td>3 5 1 7 2 6 4 5 3 4 5 3 1 7</td>
<td>0.7142</td>
</tr>
<tr>
<td>1006</td>
<td>7 2 6 4 1 7 1 7 2 6 1 7 2 6</td>
<td>0.7142</td>
</tr>
<tr>
<td>1007</td>
<td>7 2 5 3 1 7 2 6 4 5 3 1 7 6</td>
<td>0.7857</td>
</tr>
<tr>
<td>1008</td>
<td>2 6 4 7 2 6 4 5 1 7 2 6 4 5</td>
<td>0.7857</td>
</tr>
<tr>
<td>1009</td>
<td>1 7 2 6 4 5 7 4 5 7 2 6 4 5</td>
<td>0.7142</td>
</tr>
<tr>
<td>1010</td>
<td>5 3 1 7 2 6 4 1 7 1 3 1 5 6</td>
<td>0.5714</td>
</tr>
</tbody>
</table>
I. Identification

1. Program Name: PAT
2. Category: Pattern Construction
3. Purpose: Punch histograms of VARGUS 7 patterns on cards by utilizing the punch-column binary feature of the IBM 1401.
4. Date: September 26, 1968
5. Programmer: Phillip R. Jones

II. Use Information

6. Operating Instructions: Sense switch "B" off generates one column per digit; "B" on generates two columns per digit.
7. Subroutines Required: None.
8. Error Checks: ID - checks card columns 3-7 for blank.
   Input - checks for invalid characters.
III. Mathematical Method and References

None.

IV. Listing of Source Program

Attached.

V. Flow Chart

Attached.

VI. Sample Run

Not attached.

VII. Remarks

Testing has been extensive and all attempts to punch incorrect patterns have failed.
<table>
<thead>
<tr>
<th>ADDR</th>
<th>LABEL</th>
<th>TAG</th>
<th>SEQUENCE NUMBERS OF INSTRUCTIONS WHERE SYMBOL APPEARS</th>
</tr>
</thead>
<tbody>
<tr>
<td>00618</td>
<td>BEGIN</td>
<td></td>
<td>0007 0034 0078 0135 0113</td>
</tr>
<tr>
<td>01245</td>
<td>EOJ</td>
<td></td>
<td>0111 0137</td>
</tr>
<tr>
<td>01298</td>
<td>GM</td>
<td></td>
<td>0113 0035 0089 0092 0095 0097 0137 0138</td>
</tr>
<tr>
<td>00660</td>
<td>P0112</td>
<td></td>
<td>0017 0022</td>
</tr>
<tr>
<td>00698</td>
<td>P0117</td>
<td></td>
<td>0023 0013</td>
</tr>
<tr>
<td>00790</td>
<td>P0209</td>
<td></td>
<td>0035 0023</td>
</tr>
<tr>
<td>00801</td>
<td>P0211</td>
<td></td>
<td>0037 0024</td>
</tr>
<tr>
<td>00812</td>
<td>P0213</td>
<td></td>
<td>0039 0025</td>
</tr>
<tr>
<td>00823</td>
<td>P0215</td>
<td></td>
<td>0041 0026</td>
</tr>
<tr>
<td>00834</td>
<td>P0217</td>
<td></td>
<td>0043 0027</td>
</tr>
<tr>
<td>00845</td>
<td>P0219</td>
<td></td>
<td>0045 0023</td>
</tr>
<tr>
<td>00856</td>
<td>P0301</td>
<td></td>
<td>0047 0029</td>
</tr>
<tr>
<td>00867</td>
<td>P0303</td>
<td></td>
<td>0049 0030</td>
</tr>
<tr>
<td>00878</td>
<td>P0305</td>
<td></td>
<td>0051 0031</td>
</tr>
<tr>
<td>00889</td>
<td>P0307</td>
<td></td>
<td>0053 0032</td>
</tr>
<tr>
<td>00896</td>
<td>P0308</td>
<td></td>
<td>0054 0035 0038 0042 0044 0046 0048</td>
</tr>
<tr>
<td>00936</td>
<td>P0314</td>
<td></td>
<td>0060 0037 0102 0104 0155 0158</td>
</tr>
<tr>
<td>00987</td>
<td>P0401</td>
<td></td>
<td>0069 0032</td>
</tr>
<tr>
<td>01063</td>
<td>P0412</td>
<td></td>
<td>0080 0059</td>
</tr>
<tr>
<td>01081</td>
<td>P0415</td>
<td></td>
<td>0083 0070</td>
</tr>
<tr>
<td>01092</td>
<td>P0417</td>
<td></td>
<td>0085 0071</td>
</tr>
<tr>
<td>01103</td>
<td>P0419</td>
<td></td>
<td>0087 0072</td>
</tr>
<tr>
<td>01114</td>
<td>P0501</td>
<td></td>
<td>0089 0073</td>
</tr>
<tr>
<td>01132</td>
<td>P0504</td>
<td></td>
<td>0092 0074</td>
</tr>
<tr>
<td>01150</td>
<td>P0507</td>
<td></td>
<td>0095 0075</td>
</tr>
<tr>
<td>01161</td>
<td>P0509</td>
<td></td>
<td>0097 0075</td>
</tr>
<tr>
<td>01175</td>
<td>P0511</td>
<td></td>
<td>0099 0082 0084 0085 0088 0091 0094 0096</td>
</tr>
<tr>
<td>01213</td>
<td>P0517</td>
<td></td>
<td>0105 0055 0157</td>
</tr>
<tr>
<td>01218</td>
<td>P0519</td>
<td></td>
<td>0107 0007</td>
</tr>
<tr>
<td>01233</td>
<td>P0601</td>
<td></td>
<td>0109 0021 0123 0154</td>
</tr>
<tr>
<td>01299</td>
<td>P0701</td>
<td></td>
<td>0115 0051</td>
</tr>
<tr>
<td>01367</td>
<td>P0711</td>
<td></td>
<td>0125 0115</td>
</tr>
<tr>
<td>01385</td>
<td>P0714</td>
<td></td>
<td>0128 0115</td>
</tr>
<tr>
<td>01403</td>
<td>P0717</td>
<td></td>
<td>0131 0117</td>
</tr>
<tr>
<td>01421</td>
<td>P0720</td>
<td></td>
<td>0134 0113</td>
</tr>
<tr>
<td>01439</td>
<td>P0803</td>
<td></td>
<td>0137 0119</td>
</tr>
<tr>
<td>01457</td>
<td>P0806</td>
<td></td>
<td>0140 0120</td>
</tr>
<tr>
<td>01489</td>
<td>P0811</td>
<td></td>
<td>0145 0121</td>
</tr>
<tr>
<td>01507</td>
<td>P0814</td>
<td></td>
<td>0148 0122</td>
</tr>
<tr>
<td>01535</td>
<td>P0818</td>
<td></td>
<td>0152 0127 0130 0133 0135 0139 0144 0147</td>
</tr>
<tr>
<td>00600</td>
<td>START</td>
<td></td>
<td>0004 0159</td>
</tr>
<tr>
<td>00089</td>
<td>X1</td>
<td>III</td>
<td>0013 0019 0055 0058 0066 0099 0152</td>
</tr>
<tr>
<td>00094</td>
<td>X2</td>
<td>III</td>
<td>0014 0020 0055 0059 0100 0153</td>
</tr>
</tbody>
</table>
SEQ PGLIN LABEL OPCD OPERAND
0001 01020 PT219 JOB COLUMN BINARY PATTERN BUILDER VER 2 MOD 1
0002 01023 CTL 4111 S
0003 01025 ORG 600
0004 01030 START SW 1,87
0005 01040 SW 92
0006 01050 MLCWA GM,81
0007 01060 BEGIN BLC P0519 TEST LAST CARD
0008 01061 CW 201 READ A CARD
0009 01070 R
0010 01071 BIN *+5,B
0011 01072 B *+5
0012 01073 SW 201
0013 01080 SBR X1,3 CLEAR BINARY AREA
0014 01090 SBR X2,75
0015 01100 CS 599
0016 01110 CS
0017 01120 P0112 BCE *+5,0+X1, TEST FOR BLANKS
0018 01125 B P0117 BRANCH ON GOOD ID
0019 01130 SBR X1,1+X1
0020 01140 SBR X2,1+X2
0021 01150 BCE P0601,089,B TEST FOR ID LENGTH ERROR
0022 01160 B P0112
0023 01170 P0117 BCE P0209,0+X1,0 TEST FOR ZERO
0024 01180 BCE P0211,0+X1,1 ONE
0025 01190 BCE P0213,0+X1,2 TWO
0026 01200 BCE P0215,0+X1,3 THREE
0027 01201 BCE P0217,0+X1,4 FOUR
0028 01202 BCE P0219,0+X1,5 FIVE
0029 01203 BCE P0301,0+X1,6 SIX
0030 01204 BCE P0303,0+X1,7 SEVEN
0031 01205 BCE P0305,0+X1,8 EIGHT
0032 01206 BCE P0307,0+X1,9 NINE
0033 01207 WCP 1 BUILD COLUMN BINARY ZERO
0034 01208 B BEGIN
0035 01209 P0209 MLNS *8*,400+X2 ONE
0036 01210 B P0308 TWO
0037 01210 P0211 MLNS *4*,400+X2 THREE
0038 01210 B P0308 FOUR
0039 01210 P0213 MLNS *2*,400+X2 FIVE
0040 01210 B P0308 SIX
0041 01210 P0215 MLNS *1*,400+X2 SEVEN
0042 01210 B P0308 EIGHT
0043 01210 P0217 MLZS *J*,500+X2
0044 01210 B P0308
0045 01210 P0219 MLZS *Z*,500+X2
0046 01210 B P0308
0047 01210 P0301 MLNS *8*,500+X2
0048 01210 B P0308
0049 01210 P0301 MLNS *4*,500+X2
0050 01210 B P0308
0051 01210 P0301 MLNS *2*,500+X2
0052 01210 B P0308
<table>
<thead>
<tr>
<th>SEQ</th>
<th>PGLIN</th>
<th>LABEL</th>
<th>OPCD</th>
<th>OPERAND</th>
<th>A/C V1-M13</th>
</tr>
</thead>
<tbody>
<tr>
<td>0053</td>
<td>03070</td>
<td>P0307</td>
<td>MLNS</td>
<td>'1',500+X2</td>
<td></td>
</tr>
<tr>
<td>0054</td>
<td>03080</td>
<td>P0308</td>
<td>BCE</td>
<td>x19,89,7</td>
<td></td>
</tr>
<tr>
<td>0055</td>
<td>03081</td>
<td>SBR</td>
<td>X1,1+X1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0056</td>
<td>03082</td>
<td>SBR</td>
<td>X2,1+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0057</td>
<td>03090</td>
<td>B</td>
<td>P0112</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0058</td>
<td>03120</td>
<td>SBR</td>
<td>X1,14</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0059</td>
<td>03130</td>
<td>SBR</td>
<td>X2,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0060</td>
<td>03140</td>
<td>BCE</td>
<td>x13,0+X1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0061</td>
<td>03145</td>
<td>BW</td>
<td>P0701,201</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0062</td>
<td>03147</td>
<td>B</td>
<td>P0401</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0063</td>
<td>03150</td>
<td>BCE</td>
<td>x5,88,6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0064</td>
<td>03160</td>
<td>BCE</td>
<td>x9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0065</td>
<td>03170</td>
<td>BCE</td>
<td>P0517,89,7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0066</td>
<td>03180</td>
<td>SBR</td>
<td>X1,1+X1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0067</td>
<td>03190</td>
<td>B</td>
<td>P0314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0068</td>
<td>03200</td>
<td>BCE</td>
<td>x1,0+X1,0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0069</td>
<td>04010</td>
<td>P0401</td>
<td>BCE</td>
<td>P0412,0+X1,0</td>
<td></td>
</tr>
<tr>
<td>0070</td>
<td>04020</td>
<td>BCE</td>
<td>P0415,0+X1,1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0071</td>
<td>04030</td>
<td>BCE</td>
<td>P0417,0+X1,2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0072</td>
<td>04040</td>
<td>BCE</td>
<td>P0419,0+X1,3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0073</td>
<td>04050</td>
<td>BCE</td>
<td>P0501,0+X1,4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0074</td>
<td>04060</td>
<td>BCE</td>
<td>P0504,0+X1,5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0075</td>
<td>04070</td>
<td>BCE</td>
<td>P0507,0+X1,6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0076</td>
<td>04080</td>
<td>BCE</td>
<td>P0509,0+X1,7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0077</td>
<td>04090</td>
<td>WCP</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0078</td>
<td>04100</td>
<td>B</td>
<td>BEGIN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0079</td>
<td>04110</td>
<td>*</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0080</td>
<td>04120</td>
<td>P0412</td>
<td>MLC</td>
<td>x4,00+X2</td>
<td></td>
</tr>
<tr>
<td>0081</td>
<td>04130</td>
<td>MLC</td>
<td>x5,00+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0082</td>
<td>04140</td>
<td>B</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0083</td>
<td>04150</td>
<td>P0415</td>
<td>MLNS</td>
<td>'1',500+X2</td>
<td></td>
</tr>
<tr>
<td>0084</td>
<td>04160</td>
<td>B</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0085</td>
<td>04170</td>
<td>P0417</td>
<td>MLNS</td>
<td>'3',500+X2</td>
<td></td>
</tr>
<tr>
<td>0086</td>
<td>04180</td>
<td>B</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0087</td>
<td>04190</td>
<td>P0419</td>
<td>MLNS</td>
<td>'7',500+X2</td>
<td></td>
</tr>
<tr>
<td>0088</td>
<td>04200</td>
<td>B</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0089</td>
<td>05010</td>
<td>P0501</td>
<td>MLC</td>
<td>G4,500+X2</td>
<td></td>
</tr>
<tr>
<td>0090</td>
<td>05020</td>
<td>MLZS</td>
<td>x4,500+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0091</td>
<td>05030</td>
<td>MLC</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0092</td>
<td>05040</td>
<td>P0504</td>
<td>MLC</td>
<td>G4,500+X2</td>
<td></td>
</tr>
<tr>
<td>0093</td>
<td>05050</td>
<td>MLZS</td>
<td>x2,500+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0094</td>
<td>05060</td>
<td>MLC</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0095</td>
<td>05070</td>
<td>P0507</td>
<td>MLC</td>
<td>G4,500+X2</td>
<td></td>
</tr>
<tr>
<td>0096</td>
<td>05080</td>
<td>MLC</td>
<td>P0511</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0097</td>
<td>05090</td>
<td>P0509</td>
<td>MLC</td>
<td>G4,500+X2</td>
<td></td>
</tr>
<tr>
<td>0098</td>
<td>05100</td>
<td>MLNS</td>
<td>'1',400+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0099</td>
<td>05110</td>
<td>P0511</td>
<td>SBR</td>
<td>X1,1+X1</td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>05120</td>
<td>SBR</td>
<td>X2,1+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>05130</td>
<td>BCE</td>
<td>x5,88,6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0102</td>
<td>05140</td>
<td>BCE</td>
<td>P0314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0103</td>
<td>05150</td>
<td>BCE</td>
<td>x5,89,8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0104</td>
<td>05160</td>
<td>BCE</td>
<td>P0314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>PGLIN</td>
<td>LABEL</td>
<td>OPCODE</td>
<td>OPERAND</td>
<td>A/C V1-M13</td>
</tr>
<tr>
<td>-----</td>
<td>-------</td>
<td>-------</td>
<td>--------</td>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>0105</td>
<td>05170</td>
<td>P0517</td>
<td>PCB</td>
<td>BEGIN</td>
<td></td>
</tr>
<tr>
<td>0106</td>
<td>05180</td>
<td></td>
<td></td>
<td></td>
<td>PUNCH COLUMN BINARY</td>
</tr>
<tr>
<td>0107</td>
<td>05190</td>
<td>P0519</td>
<td>WCP</td>
<td>EDJ</td>
<td></td>
</tr>
<tr>
<td>0108</td>
<td>05200</td>
<td></td>
<td>M</td>
<td>999,999</td>
<td></td>
</tr>
<tr>
<td>0109</td>
<td>06010</td>
<td>P0601</td>
<td>WCP</td>
<td>1</td>
<td>DATA CARD ERROR</td>
</tr>
<tr>
<td>0110</td>
<td>06020</td>
<td></td>
<td>B</td>
<td>BEGIN</td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>06030</td>
<td>EDJ</td>
<td>DCW</td>
<td>&quot;END OF JOB -- REMOVE ANY CORRECTIONS TYPED ON&quot;</td>
<td></td>
</tr>
<tr>
<td>0112</td>
<td>06040</td>
<td></td>
<td></td>
<td>&quot; CONSOLE&quot;</td>
<td></td>
</tr>
<tr>
<td>0113</td>
<td>06050</td>
<td>GM</td>
<td>DCW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0114</td>
<td>07000</td>
<td></td>
<td></td>
<td></td>
<td>DOUBLED RAW OPTION</td>
</tr>
<tr>
<td>0115</td>
<td>07010</td>
<td>P0701</td>
<td>BCE</td>
<td>P0711,0+X1,0</td>
<td>TEST FOR ZERO</td>
</tr>
<tr>
<td>0116</td>
<td>07020</td>
<td>BCE</td>
<td>P0714,0+X1,1</td>
<td>ONE</td>
<td></td>
</tr>
<tr>
<td>0117</td>
<td>07030</td>
<td>BCE</td>
<td>P0717,0+X1,2</td>
<td>TWO</td>
<td></td>
</tr>
<tr>
<td>0118</td>
<td>07040</td>
<td>BCE</td>
<td>P0720,0+X1,3</td>
<td>THREE</td>
<td></td>
</tr>
<tr>
<td>0119</td>
<td>07050</td>
<td>BCE</td>
<td>P0803,0+X1,4</td>
<td>FOJ</td>
<td></td>
</tr>
<tr>
<td>0120</td>
<td>07060</td>
<td>BCE</td>
<td>P0805,0+X1,5</td>
<td>FIVE</td>
<td></td>
</tr>
<tr>
<td>0121</td>
<td>07070</td>
<td>BCE</td>
<td>P0811,0+X1,6</td>
<td>SIX</td>
<td></td>
</tr>
<tr>
<td>0122</td>
<td>07080</td>
<td>BCE</td>
<td>P0814,0+X1,7</td>
<td>SEVEN</td>
<td></td>
</tr>
<tr>
<td>0123</td>
<td>07090</td>
<td>B</td>
<td>P0601</td>
<td></td>
<td>DATA ERROR</td>
</tr>
<tr>
<td>0124</td>
<td>07100</td>
<td></td>
<td></td>
<td>BUILD DOUBLED BINARY COLUMN</td>
<td></td>
</tr>
<tr>
<td>0125</td>
<td>07110</td>
<td>P0711</td>
<td>MLC</td>
<td>'401+X2</td>
<td>ZERO (NO BITS)</td>
</tr>
<tr>
<td>0126</td>
<td>07120</td>
<td>MLC</td>
<td>'501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0127</td>
<td>07130</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0128</td>
<td>07140</td>
<td>P0714</td>
<td>MLNS</td>
<td>'1',500+X2</td>
<td>ONE (1-BIT)</td>
</tr>
<tr>
<td>0129</td>
<td>07150</td>
<td>MLNS</td>
<td>'1',501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0130</td>
<td>07160</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0131</td>
<td>07170</td>
<td>P0717</td>
<td>MLNS</td>
<td>'3',500+X2</td>
<td>TWO (1,2-BITS)</td>
</tr>
<tr>
<td>0132</td>
<td>07180</td>
<td>MLNS</td>
<td>'3',501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0133</td>
<td>07190</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0134</td>
<td>07200</td>
<td>P0720</td>
<td>MLNS</td>
<td>'7',500+X2</td>
<td>THREE (1,2,4-BITS)</td>
</tr>
<tr>
<td>0135</td>
<td>08010</td>
<td>MLNS</td>
<td>'7',501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0136</td>
<td>08020</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0137</td>
<td>08030</td>
<td>P0803</td>
<td>MLNS</td>
<td>GM,500+X2</td>
<td>FOUR (1,2,4,8-BITS)</td>
</tr>
<tr>
<td>0138</td>
<td>08040</td>
<td>MLNS</td>
<td>GM,501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0139</td>
<td>08050</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0140</td>
<td>08060</td>
<td>P0806</td>
<td>MLNS</td>
<td>GM,500+X2</td>
<td>FIVE (1,2,4,8,A-BITS)</td>
</tr>
<tr>
<td>0141</td>
<td>08070</td>
<td>MLZS</td>
<td>'Z',500+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0142</td>
<td>08080</td>
<td>MLNS</td>
<td>GM,501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0143</td>
<td>08090</td>
<td>MLZS</td>
<td>'Z',501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0144</td>
<td>08100</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0145</td>
<td>08110</td>
<td>P0811</td>
<td>MLC</td>
<td>GM,500+X2</td>
<td>SIX (1,2,4,8,A,B-BITS)</td>
</tr>
<tr>
<td>0146</td>
<td>08120</td>
<td>MLC</td>
<td>GM,501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0147</td>
<td>08130</td>
<td>B</td>
<td>P0818</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0148</td>
<td>08140</td>
<td>P0814</td>
<td>MLC</td>
<td>GM,500+X2</td>
<td>SEVEN (1,2,4,8,A,B,1-BITS)</td>
</tr>
<tr>
<td>0149</td>
<td>08150</td>
<td>MLC</td>
<td>GM,501+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0150</td>
<td>08160</td>
<td>MLNS</td>
<td>'1',400+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0151</td>
<td>08170</td>
<td>MLNS</td>
<td>'1',401+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0152</td>
<td>08180</td>
<td>P0818</td>
<td>SBR</td>
<td>X1,1+X1</td>
<td></td>
</tr>
<tr>
<td>0153</td>
<td>08190</td>
<td>SBR</td>
<td>X2,2+X2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0154</td>
<td>08200</td>
<td>BCE</td>
<td>P0601,93,8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0155</td>
<td>08210</td>
<td>BCE</td>
<td>*+5,88,6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0156</td>
<td>08220</td>
<td>B</td>
<td>P0314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SEQ</td>
<td>PGLIN LABEL</td>
<td>OPCD</td>
<td>OPERAND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>-------------</td>
<td>-------</td>
<td>-------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0157</td>
<td>08230</td>
<td>BCE</td>
<td>P0517, 89, 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0158</td>
<td>09010</td>
<td>B</td>
<td>P0314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0159</td>
<td>09020</td>
<td>END</td>
<td>START</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0160</td>
<td></td>
<td>LTRL</td>
<td>'8'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0161</td>
<td></td>
<td>LTRL</td>
<td>'4'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0162</td>
<td></td>
<td>LTRL</td>
<td>'2'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0163</td>
<td></td>
<td>LTRL</td>
<td>'1'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0164</td>
<td></td>
<td>LTRL</td>
<td>'J'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0165</td>
<td></td>
<td>LTRL</td>
<td>'2'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0166</td>
<td></td>
<td>LTRL</td>
<td>''</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0167</td>
<td></td>
<td>LTRL</td>
<td>'3'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0168</td>
<td></td>
<td>LTRL</td>
<td>'7'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0169</td>
<td></td>
<td>LTRL</td>
<td>''</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

END OF LISTING
START
1 House-keeping

BEGIN
2 Has last card been read? Yes
   Print E0J on console
   Set up I.D. digits in column binary in punch area
   Reset pattern column and data column counters

3 Reset double column switch
   Halt 999
   Set up column in punch area according to digit indicated by data column counter

4 Read a card
   Is pattern in doubled column? No
   Move binary column to next two positions in punch area according to counter
   Add '1' to positions in punch area according to counter

5 Is pattern in doubled column? Yes
   Turn 'on' double column switch
   Reset I.D. column indicator
6

7 Has I.D. blank? Yes
   Print data card on console
   Add '2' to data column counter
   Has last data digit been processed? No
   Punch column binary
   C3

8

9 C1 A2
I. Identification

1. Program Name: VARGUS 9
2. Category: Pattern Generation
3. Purpose: To compute patterns by producing probabilistic variations on a prototype.
4. Date: January, 1966
5. Programmer: Selby Evans

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Up to 32 columns in a pattern.
4. Input data:

   The initial call to RANDY causes a START card to be read by the subroutine. See RANDY documentation for details.

   NCOL: Number of columns in a pattern, NCOL less than or equal to 32.

   NSCHM: The schema identification number. Typically it is 3 or 4 digits ending in 00.

   NINST: The number of patterns to be generated. NINST less than or equal to 99.

   MAX: A vector the first NCOL elements of which form the schema.
P: The probability matrix the $I$-th column of which controls the frequency distribution (dispersion) of the $I$-th elements of the patterns about the $I$-th schema element. NCOL such probability vectors are read in; i.e., one for each element of the schema.

5. Output Data:

   NPAT: The individual pattern identification number.
   INST: The pattern itself.
   PR: The cumulative product of the probability of each actual selection in constructing a pattern.
   FMS: The mean squared deviation of each pattern from its schema (elsewhere called pattern variance).

6. Subroutines Required:

   RANDY: A pseudo-random number generator.

III. Mathematical Method and References


IV. Listing of Source Program

Attached.
V. Sample Run

Attached.

VI. Remarks

None.
C VARGUS 9
   DIMENSION P(9,32),MAX(32),INST(32)
   CALL RANDY(R)
   WRITE(3,901)
   901 FORMAT(1,T19,'SAMPLE VARGUS NINE PATTERNS')
   WRITE(3,911)
   911 FORMAT(1,T6,'ID',T26,'PATTERN',T48,'PROB',T55,'PATTERN VARIANCE 1')
   C READ PARAMETERS
   READ(2,1):NCOL,NSCHM,NINST
   1 FORMAT(515)
      DO 101 IN=1,32
         MAX(IN)=0
      101 INST(IN)=0
      C READ PROBABILITY MATRIX
      DO 55 J=1,NCOL
      55 READ(2,2):MAX(J),(P(I,J),I=1,9)
   2 FORMAT(5I3,9F8.4)
   C FNCOL =NCOL
   C OUTPUT SCHEMA
   WRITE(3,5) NSCHM,(MAX(I),I=1,NCOL)
   5 FORMAT(3X,5,8X,32I2)
   C SET UP INSTANCE LOOP
   DO 99 IN=1,NINST
      FMS=0
      PR=1.0
      SS=0
   99 CONTINUE
   C SET UP COLUMN LOOP
   DO 89 ICOL=1,NCOL
      CALL RANDY(R)
   C CHOOSE ELEMENT
   DO 79 L=1,9
      R = R - P(L,ICOL)
      IF(R)00,79,79
   79 CONTINUE
   L=9
   80 CONTINUE
      PR=PR*P(L,ICOL)
      T= MAX(ICOL)-L
      SS = SS + T*T
   89 INST(ICOL)=L
C END INSTANCE LOOP
NPAT = NSCHM + IN
FMS = SS/FLD
WRITE(3,15) NPAT,(INST(I),I=1,NCOL),PR,FMS
15 FORMAT(3X,15,8X,14I2,2F8.4)
99 CONTINUE
CALL EXIT
END

VARIABLE ALLOCATIONS
P(R )=023E-0000 R(R )=0240 FNCOL(R )=0242 FMS(R )=0244
T(R )=024A MAX(I )=0258-024G INST(I )=0258-026C NCOL(I )=026C
IN(I )=028F J(I )=0290 I(I )=0291 ICOL(I )=0292

STATEMENT ALLOCATIONS
901 =02A0 911 =02B3 1 =02CF 2 =02D2 5 =02D7 15 =02D0 10
89 =03BE 99 =03F6

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CALLED SUBPROGRAMS
RANDY FADD FSUBX FMXY FMXYX FDIV FLD FSTO FLOAT IS'
MIOIX MIOF MIOI SUBSC FCHRI TYPEN HOLEB MAGT PRNTN E61

REAL CONSTANTS
*1000000 E01=02D8

INTEGER CONSTANTS
3=029A 2=029B 1=029C 32=029D 0=029E 9=029F

CORE REQUIREMENTS FOR ABBA
COMMON 0 INSkel COMMON 0
VARIABLES 664 PROGRAM 362

END OF COMPILATION
### Sample Vargus Nine Patterns

<table>
<thead>
<tr>
<th>ID</th>
<th>Pattern</th>
<th>Prob.</th>
<th>Pattern Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>43574654565575</td>
<td>0.0000</td>
<td>0.5714</td>
</tr>
<tr>
<td>2001</td>
<td>43574654565553</td>
<td>0.0001</td>
<td>0.3571</td>
</tr>
<tr>
<td>2002</td>
<td>43572654575575</td>
<td>0.0000</td>
<td>0.4285</td>
</tr>
<tr>
<td>2003</td>
<td>23574653565475</td>
<td>0.0000</td>
<td>0.7142</td>
</tr>
<tr>
<td>2004</td>
<td>43574654566575</td>
<td>0.0055</td>
<td>0.0714</td>
</tr>
<tr>
<td>2005</td>
<td>43684654565593</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2006</td>
<td>43574654565575</td>
<td>0.0870</td>
<td>0.0000</td>
</tr>
<tr>
<td>2007</td>
<td>43573654565575</td>
<td>0.0055</td>
<td>0.0714</td>
</tr>
<tr>
<td>2008</td>
<td>43575644565575</td>
<td>0.0003</td>
<td>0.1428</td>
</tr>
<tr>
<td>2009</td>
<td>43476654665575</td>
<td>0.0000</td>
<td>0.4285</td>
</tr>
<tr>
<td>2010</td>
<td>43574654565574</td>
<td>0.0055</td>
<td>0.0714</td>
</tr>
</tbody>
</table>
I. Identification

1. Program Name: VARG 9
2. Category: Pattern Generation
3. Purpose: Compute VARGUS 9 patterns and print dots that can be connected to form a serriform VARGUS 9 pattern.
4. Date: April, 1970
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Maximum of 32 element patterns.
4. Input Data:
   
   First call to subroutine RANDY causes START card to be read. See RANDY documentation for details.

   IMØ: Month.
   IDS: Day.
   IYR: Year.
   USER: User information; e.g., name.
   NCØL: Number of digits per pattern.
   NSHEM: Identification number of schema. Should be 3 digits at most.
   NINST: The number of patterns desired.
   MAX: A vector the first NCØL elements of which contain the schema.
PRØB: A probability matrix the I-th column of which contains probabilities which control the frequency distribution of values of the I-th element in the patterns produced. For example if MAX(I) is a 5, then the column PROB (1,I) to PROB (32,I) might be 0,0,0.1, 0.1, 0.6, 0.1, 0.1, 0,0,0, etc.

5. Output Data:

ARY: A matrix containing alphameric blanks and points. The points may be connected to form a serriform pattern.

NPAT: Pattern identification number.

INST: The vector of digits which constitute the basic pattern.

PR: A cumulative product of probabilities of the elements actually chosen.

FMS: The mean square deviations of the pattern from its prototype. Also called pattern variance.

All output to logical unit 8 is to produce a tape (in binary) utilized by program DVAR.

6. Subroutines Required:

RANDY: A pseudo-random number generator.

III. Mathematical Methods and References

IV. Listing of Source Program

    Attached.

V. Sample Run

    Attached.

VI. Remarks

    None.
DIMENSION PROB (9,32), MAX(32), INST(32), USER(3)
DIMENSION ARY(20,32)
DATA FK)INTs BLANK /'s't 0 '/
RR = 1.
CALL RANDDY(RR)
1 FORMAT (315)
2 FORMAT (15,3X ,9F8.5)
3 FORMAT (313,3A4)
4 FORMAT (9H1VARGUS 9, 313, 3A4, 4X, 6HSHEMA, I5, 1H,,I5, 5H INST)
5 FORMAT (6H0 VR9 , I4, 1H, 8(1H ,411))
6 FORMAT (6H0 VR9 , I4, 1H, 8(1H ,411)/ E14.8, F12.5)
7 FORMAT (1H1)
8 FORMAT (50X, 14A3//)
9 FORMAT (////////////)
RENEW 8
C
READ DATA AND USER
8000 READ (2,3) IMOD, IDA, IYR, (USER(I), I=1,3)
C
READ PARAMETERS
READ (2,1) NCOL, NSHEM, NINST
DO 101 IN=1,32
MAX(IN)=0
101 INST(IN)=0
C
READ PROBABILITY MATRIX
READ (2,2) (MAX(J), (PROB(I,J), I=1,9), J=1,NCOL)
C
FNCOL=NCOL
C
OUTPUT IDENTIFICATION
WRITE (3,4) IMOD, IDA, IYR, (USER(I), I=1,3), NSHEM, NINST
WRITE (8) IMOD, IDA, IYR, USER
WRITE (8) :NCOL, NSHEM, NINST
C
OUTPUT SCHEMA
WRITE (3,5) NSHEM, (MAX(I), I=1,NCOL)
WRITE (8) :NSHEM, (MAX(I), I=1,NCOL)
DO 10 I=1,20
DO 10 J=1,32
10 ARY(I,J)=BLANK
IP=10
ARY(IP+1)=POIN
ARY(IP,NCOL)=POIN
DO 15 I=1,NCOL
II=10-MAX(I)
15 ARY(II,I)=POIN
WRITE (3,9)
WRITE (3,8) ((ARY(I,J), J=1,14), I=1,IP)
WRITE (3,7)
C
SET INSTANCE LOOP
DO 99 I=1,NINST
PR=1.0
SS=0.0
DO 20 I=1,20
DO 20 J=1,32
20 ARY(I,J)=BLANK
ARY(IP,NCOL)=POINT
ARY(IP,1)=POINT
C
SET COLUMN LOOP
DO 89 ICOL=1,NCOL
CALL RANDY(RR)
R=RR
C
CHOOSE ELEMENT
DO 79 L=1,9
R=R-PROB(L,ICOL)
IF(R)80,79,79
79 CONTINUE
L=9
80 CONTINUE
PR=PR+PROB(L,ICOL)
T=MAX(ICOL)-L
SS=SS+(T*T)
89 INST(ICOL)=L
NPAT=NSHEM+IN
FMS=SS/NCOL
WRITE (3,6) NPAT, (INST(I), I=1,32), PR, FMS
WRITE (8) NPAT, (INST(I), I=1,NCOL), PR, FMS
DO 30 J=1,NCOL
II=10-INST(J)
30 ARY(I,J)=POINT
WRITE (3,9)
WRITE (3,8) ((ARY(I,J), J=1,14), I=1,IP)
WRITE (3,7)
99 CONTINUE
C
END INSTANCE LOOP
CALL EXIT
END

VARIABLE ALLOCATIONS
PROB(R)=023E-0000 USER(R)=0244-0240 ARY(R)=0744-0246 RR(R)=0746
POINT(R)=074C PR(R)=074E SS(R)=0750 R(R)=0752
MAX(I)=0777-0758 INST(I)=0797-0778 IMO(I)=0798 IDA(I)=0799
NCOL(I)=079C NSHEM(I)=079D NINST(I)=079E IN(I)=079F
II(I)=07A2 ICOL(I)=07A3 L(I)=07A4 NPAT(I)=07A5

UNREFERENCED STATEMENTS
STATEMENT ALLOCATIONS

1 =07B6  2 =07B9  3 =07BE  4 =07C3  5 =07DB  6 =07E9  7
101 =0848  10 =08F7  15 =0938  20 =0990  79 =09E2  80 =09EF  89

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IDCS

CALLED SUBPROGRAMS
RANDY  FADD  FSUBX  FMPLY  FMPLYX  FDIV  FLD  FSTD  FSTOX  FLI
MWRT  MCMP  MIOFX  MIOIX  MIOF  MIDI  SUBSC  REWND  UIDAF  UII
PRNTN  EBPRT  CARON

REAL CONSTANTS
  .1E000000E 01=07AB  .000000E 00=07AA

INTEGER CONSTANTS
  8=07AC  2=07AD  1=07AE  3=07AF  32=07B0  0=07B1

CORE REQUIREMENTS FOR VARG
COMMON  0 INSKEH COMMON  0
VARIABLES  1960  PROGRAM  792

END OF COMPILATION
VR9 2004, 4357 4654 5665 7500 0000 0000 0000 0000
• 55253123E-02 0.07142
I. Identification

1. Program Name: DVAR
2. Category: Pattern Measurement and Description
3. Purpose: Finds Pythagorean distances of a set of patterns from a given pattern.
4. Date: April, 1970
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: The distance of each pattern in a set may be found from up to 4 defined patterns.
4. Input data:
   
   Tape produced by program VARG 9.
   
   TITLE: Distance values against which to compare computed distances and thereby form a histogram.
   
   NPRØT: Number of schema patterns: up to four.

5. Output data:
   
   NPAT: Pattern identification number.
   
   DIST: Distance of a pattern from the set from NPRØT given patterns.
   
   AVER: Average pattern variance; i.e., average of FMS values.
ARY: Histogram of dots.
TITLE: Abscissa of the histogram.
IFREQ: A frequency count of the dots printed over each abscissa value.
NØS: Histogram corresponding to ARY but constructed from actual pattern numbers not dots.

6. Subroutines Required:
None.

III. Mathematical Methods and References

IV. Listing of Source Program
Attached.

V. Sample Run
Attached.

VI. Remarks
None.
// JOB 30 JUN 71 13.650 HRS
// U-2050-003
// FOR DVAR 30 JUN 71 13.650 HRS
*LIST
*IDCS (CARD, 1443 PRINTER, DISK, PLOTTER, MAGNETIC TAPE)
*ONE WORD INTEGERS
**NONPROCESS PROGRAM
C

DVAR

DIMENSION USER(3), MAX(32), INST(32), ARY(24, 16), IFREQ(24),
1 TITLE(24), NOS(24, 16), NPROT(14, 4), DIST(4)

DATA (INT, HLANK /* , */
1 FORMAT(9H1VARUS 9, 3I3, 3A4, 4X, 6HSHEM, A15, 1H,, 15, 5H INST)
2 FORMAT(13H AVERAGE PV = ,F12.5)
3 FORMAT(24A5)
4 FORMAT(16O, 14, 23I5)
5 FORMAT(1X, 4F4.2, 23F5.2//)
6 FORMAT(8F10.5)
7 FORMAT(5X, 4(10X, 14I1))
8 FORMAT(15, 4(10X, F14.5))
9 FORMAT(4(5X, 14I1))
10 FORMAT(15, 4F15.5)

REWRITE 8

C

READ(2,6) TITLE
READ(2,9) NPROT
READ(8) MD, IDA, IYR, USER
READ(8) NCOL, NSHEM, NINST
WRITE(3,1) MD, IDA, IYR, (USER(I), I=1,3), NSHEM, NINST
WRITE(3,7) NPROT
READ(8) NSHEM, (MAX(I), I=1, NCOL)
DO 14 L=1,4
TEMP=0.
DO 13 KK=1, NCOL
TEM=NPROT(KK, L) - MAX(KK)
13 TEMP=TEMP+TEM*TEM
14 DIST(L)=SORT(TEMP)
WRITE(3,8) NSHEM, DIST

C

DO 20 I=1, 24
IFREQ(I)=0
NPROT(I)=1
DO 20 J=1, 16
NOS(I,J)=0
20 ARY(I, J)=BLANK

C

SUM=0.
DO 50 I=1, NINST
READ(3) NPA1, (INST(K), K=1, NCOL), PR, FMS
SUM=SUM+FMS
DO 16 L=1, 4
TEMP=0.
DO 15 KK=1, NCOL
TEM=NPROT(KK, L) - INST(KK)
15 TEMP=TEMP+TEM*TEM
16 DIST(L)=SORT(TEMP)
      WRITE(3,8) NPAT,DIST

C
   DO 40 J=1,24
      IF(FMS-TITLE(J)) 30,30,40
30   KK=NPONT(J)
      NPONT(J)=NPONT(J)+1
      IFREQ(J)=IFREQ(J)+1
   GO TO 50
40  CONTINUE
50  CONTINUE
   AVER=SUM(ARRAY(NINST))
      WRITE(3,2) AVER
   DO 70 I=1,24
   DO 70 J=1,16
   IF(IFREQ(I)-J) 70,60,60
50  CONTINUE
   AVER=SUM(ARRAY(NINST))
      WRITE(3,2) AVER
   DO 70 I=1,24
   DO 70 J=1,16
   IF(IFREQ(I)-J) 70,60,60
50  CONTINUE
   AVER=SUM(ARRAY(NINST))
      WRITE(3,2) AVER
60  CONTINUE
   AVER=SUM(ARRAY(NINST))
      WRITE(3,2) AVER

VARIABLE ALLOCATIONS
USER(R )=0004-0006  ARY(R )=0304-0006  TITLE(R )=0334-0306  DIST(R )=033C-
BLANK(R )=0342
MAX(I )=0360-034E  INST(I )=0390-036E  IFREQ(I )=03A5-038E  NDS(I )=0325-
IMO(I )=0576  ID(I )=0577  IYR(I )=0578  NCUL(I )=0579
I1(I )=057C  L(I )=057D  KK(I )=057E  J(I )=057F
IMAX(I )=0582  IB(I )=0583  II(I )=0584  IX(I )=0585

UNREFERENCED STATEMENTS
10

STATEMENT ALLOCATIONS
1  =0592  2  =05AA  3  =05A4  4  =05A7  5  =0580  6  =05C4  7

56
865 =05DE 13 =0669 14 =067A 20 =068A 15 =0724 16 =0735 30
70 =07C6 52 =0800 53 =0809 46 =0822

FEATURES SUPPORTED
MINIPROCESS
ONE WORD INTEGERS
INCS

CALLED SUBPROGRAMS
FSQRT FADD FSUBX FMPY FLD FSTD FSTOX FDVR FLOAT IS
XCOMP MIDAI MIOAF M IOFX M IOIX MIOF MIOI SUBSC REWND IB

REAL CONSTANTS
.000000E 00 =058B

INTEGER CONSTANTS:
8=058A 2=058B 3=059C 1=058D 4=058E 24=058F

CORE REQUIREMENTS FOR DVAR
COMMON o INSKEL COMMON o
VARIABLES 1416 PROGRAM 716

END OF COMPILATION
<table>
<thead>
<tr>
<th>Year</th>
<th>VARGUS 9 5 26 71 I.S.C.S.</th>
<th>SCHEMA 2000</th>
<th>30 INST</th>
<th>4574373737365</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2001</td>
<td>2.82942</td>
<td>7.48331</td>
<td>7.21110</td>
<td>7.41619</td>
</tr>
<tr>
<td>2002</td>
<td>2.23606</td>
<td>8.06225</td>
<td>8.30662</td>
<td>8.36660</td>
</tr>
<tr>
<td>2003</td>
<td>2.44949</td>
<td>7.74846</td>
<td>7.74596</td>
<td>8.77496</td>
</tr>
<tr>
<td>2004</td>
<td>1.00000</td>
<td>7.41619</td>
<td>7.14142</td>
<td>7.74596</td>
</tr>
<tr>
<td>2005</td>
<td>3.16227</td>
<td>8.36660</td>
<td>8.71779</td>
<td>9.53939</td>
</tr>
<tr>
<td>2006</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2007</td>
<td>1.00000</td>
<td>7.41619</td>
<td>7.68114</td>
<td>7.87400</td>
</tr>
<tr>
<td>2008</td>
<td>1.41421</td>
<td>6.92820</td>
<td>6.6325</td>
<td>7.93725</td>
</tr>
<tr>
<td>2009</td>
<td>2.44949</td>
<td>7.34846</td>
<td>6.92820</td>
<td>8.88819</td>
</tr>
<tr>
<td>2010</td>
<td>1.00000</td>
<td>7.28011</td>
<td>7.28011</td>
<td>8.00000</td>
</tr>
<tr>
<td>2011</td>
<td>2.64975</td>
<td>8.18535</td>
<td>8.66025</td>
<td>8.60232</td>
</tr>
<tr>
<td>2012</td>
<td>1.41421</td>
<td>7.21110</td>
<td>6.92820</td>
<td>8.18535</td>
</tr>
<tr>
<td>2013</td>
<td>3.46410</td>
<td>7.74596</td>
<td>9.38083</td>
<td>10.34408</td>
</tr>
<tr>
<td>2014</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2015</td>
<td>1.73205</td>
<td>6.85565</td>
<td>7.14142</td>
<td>8.48528</td>
</tr>
<tr>
<td>2016</td>
<td>2.44949</td>
<td>8.36660</td>
<td>7.87400</td>
<td>9.32737</td>
</tr>
<tr>
<td>2017</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2018</td>
<td>2.00000</td>
<td>7.61577</td>
<td>7.21110</td>
<td>7.81025</td>
</tr>
<tr>
<td>2019</td>
<td>2.00000</td>
<td>8.00000</td>
<td>7.74596</td>
<td>7.93725</td>
</tr>
<tr>
<td>2020</td>
<td>1.00000</td>
<td>6.70820</td>
<td>7.00000</td>
<td>7.61577</td>
</tr>
<tr>
<td>2021</td>
<td>1.73205</td>
<td>6.24499</td>
<td>6.85565</td>
<td>7.07106</td>
</tr>
<tr>
<td>2022</td>
<td>2.44949</td>
<td>7.48331</td>
<td>6.32455</td>
<td>7.28011</td>
</tr>
<tr>
<td>2023</td>
<td>1.73205</td>
<td>7.68114</td>
<td>7.00000</td>
<td>8.48528</td>
</tr>
<tr>
<td>2024</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2025</td>
<td>2.00000</td>
<td>7.21110</td>
<td>6.63325</td>
<td>8.42615</td>
</tr>
<tr>
<td>2026</td>
<td>1.00000</td>
<td>6.70820</td>
<td>7.00000</td>
<td>7.61577</td>
</tr>
<tr>
<td>2027</td>
<td>0.00000</td>
<td>7.21110</td>
<td>7.21110</td>
<td>7.93725</td>
</tr>
<tr>
<td>2028</td>
<td>1.00000</td>
<td>7.00000</td>
<td>7.28011</td>
<td>7.61577</td>
</tr>
<tr>
<td>2029</td>
<td>1.41421</td>
<td>7.34846</td>
<td>7.48331</td>
<td>8.42615</td>
</tr>
<tr>
<td>2030</td>
<td>2.44975</td>
<td>7.00000</td>
<td>7.81025</td>
<td>8.71779</td>
</tr>
</tbody>
</table>

**AVERAGE PV = 0.25238**

0.00 0.07 0.14 0.21 0.28 0.35 0.42 0.50 0.57 0.64 0.71 0.78 0.85 0.92 1.00
I. Identification

1. Program Name: VARGUS 10A
2. Category: Pattern Generator
3. Purpose: Input basic parameters, identification information, and matrices to be used in the generation of VARGUS 10-type patterns. See Subroutine J@N for details of generation.
4. Date: July, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: The final pattern matrix is a 48 x 48 matrix. Each of the corner and side submatrices are effectively 16 x 16, but each is represented internally by a string of subscript pairs not by a complete matrix. The first subscript of each pair is considered to be a row index and the second a column index. All corner matrices begin at (16, 10) and end at (10, 16). All side matrices begin at (10, 1) and end at (10, 16).
4. Input Data:
   a. A random-number generator START card under format (17I4,4I1). See subroutine RANDY documentation for further details about the START card.
   b. IDA, N®PAT (4I3). Day, month and year in numbers, followed by the number of patterns to be generated.
   c. NH, NL (2I3). Number of side matrices and number of corner matrices, respectively.
   d. MATH (25I3). This matrix contains in each row the side submatrices in subscript form. Four cards are required for each of the NH side submatrices. See Table in Remarks.
   e. MATL (25I3). This matrix contains in each row the corner submatrices in subscript form. Four cards are required for each of the NL corner submatrices.

5. Output Data: (To printer). See subroutine PAT®T for details.

   J®IN: Selects, positions and rotates side and corner matrices.
III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

The following information is a list of the 40 side submatrices and 34 corner submatrices in subscript form. The side submatrices are identified by a leading H in the identification code and the corner submatrices by a leading L.

H1: (10,1) (9,1) (8,1) (7,2) (6,2) (5,3) (4,3) (3,3) (4,4) (4,5) (5,6) (5,7) (6,7) (6,8) (7,9) (7,10) (8,11) (8,12) (9,13) (9,14) (10,15) (10,16)
H2: (10,1) (9,1) (8,1) (7,2) (6,2) (5,3) (4,3) (3,4) (4,5) (5,5) (6,6) (7,6) (8,7) (9,7) (10,8) (11,8) (12,9) (13,9) (14,10) (15,11) (16,11) (16,12) (15,12) (15,11) (14,12) (13,13) (13,14) (12,14) (12,15) (11,15) (10,16)
H3: (10,1) (11,1) (11,2) (12,3) (12,4) (13,5) (14,5) (14,6) (15,7) (16,8) (16,9) (15,10) (15,11) (14,11) (14,12) (13,13) (13,14) (12,15) (11,16) (10,16)
H4: (10,1) (9,2) (9,3) (8,3) (8,4) (7,5) (6,6) (6,7) (5,8) (5,9) (4,9) (4,10) (3,11) (4,12) (5,13) (6,13) (7,14) (8,15) (9,15) (9,16) (10,16)
H5: (10,1) (10,2) (9,2) (8,2) (7,3) (6,4) (5,5) (4,6) (4,7) (3,8) (3,9) (3,10) (4,11) (4,12) (5,13) (6,13) (7,13) (8,13) (9,13) (10,14) (10,15) (10,16)
H6: (10,1) (10,2) (10,3) (11,4) (12,5) (12,6) (12,7) (12,8) (11,8) (10,8) (9,8) (9,9) (9,10) (9,11) (9,12) (9,13) (9,14) (9,15) (9,16) (10,16)
<table>
<thead>
<tr>
<th></th>
<th>L1</th>
<th>L2</th>
<th>L3</th>
<th>L4</th>
<th>L5</th>
<th>L6</th>
<th>L7</th>
<th>L8</th>
<th>L9</th>
<th>L10</th>
<th>L11</th>
<th>L12</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10) (9,10) (8,10) (7,10) (6,10) (5,11) (4,12) (4,13) (4,14) (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)</td>
<td>(16,10) (16,9) (15,8) (15,7) (14,6) (13,5) (12,4) (11,4) (10,3) (9,3) (8,3) (7,3) (6,4) (5,4) (4,5) (4,6) (3,7) (3,8) (3,9) (3,10) (3,11) (4,12) (5,13) (6,14) (7,15) (8,16) (9,16) (10,16)</td>
<td>(16,10) (15,11) (14,11) (13,11) (12,10) (12,9) (12,8) (12,7) (12,6) (12,5) (12,4) (11,4) (10,4) (9,4) (8,4) (7,4) (6,4) (5,4) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15) (5,15) (6,15) (7,15) (8,15) (9,16) (10,16)</td>
<td>(16,10) (16,9) (15,8) (15,7) (15,6) (15,5) (15,4) (14,4) (13,4) (12,4) (11,4) (10,4) (9,3) (8,3) (7,3) (6,3) (5,3) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15) (5,15) (6,15) (7,15) (8,15) (9,16) (10,16)</td>
<td>(16,10) (16,11) (16,12) (16,13) (15,13) (15,14) (14,15) (13,15) (13,16) (12,16) (11,16) (10,16)</td>
<td>(16,10) (16,9) (15,8) (14,7) (13,6) (12,5) (12,4) (11,3) (10,4) (9,5) (8,6) (7,6) (6,7) (5,8) (4,9) (3,10) (4,11) (5,12) (6,13) (6,14) (7,15) (8,16) (9,16) (10,16)</td>
<td>(16,10) (15,9) (15,10) (15,8) (15,7) (15,6) (14,5) (13,5) (12,5) (11,5) (10,5) (9,5) (9,6) (8,6) (7,6) (7,7) (6,7) (5,8) (5,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15) (5,15) (5,16) (6,16) (7,16) (8,16) (9,16) (10,16)</td>
<td>(16,10) (16,9) (16,8) (16,7) (15,6) (15,5) (15,4) (14,4) (13,3) (12,3) (11,3) (10,3) (9,3) (8,3) (7,3) (6,3) (5,4) (4,4) (4,5) (4,6) (3,7) (3,8) (3,9) (3,10) (3,11) (3,12) (3,13) (3,14) (4,15) (5,15) (6,15) (7,15) (8,15) (9,15) (10,16)</td>
<td>(16,10) (15,11) (14,11) (13,12) (12,12) (11,13) (10,14) (10,15) (10,16)</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (11,11) (11,12) (10,13) (10,14) (10,15) (10,16)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L13:</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (12,9) (12,8) (12,7) (12,6) (12,5) (12,4) (11,4) (10,4) (9,4) (8,4) (7,4) (6,4) (5,4) (4,4) (4,5) (4,6) (4,7) (4,8) (4,9) (4,10) (4,11) (4,12) (4,13) (4,14) (6,15) (5,15) (7,14) (8,14) (9,14) (10,14) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L14:</td>
<td>(16,10) (15,11) (14,12) (13,12) (12,13) (11,14) (11,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L15:</td>
<td>(16,10) (16,11) (15,12) (14,12) (13,13) (12,13) (11,13) (10,13) (9,13) (9,14) (9,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L16:</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10) (9,10) (8,10) (7,10) (6,10) (5,11) (5,12) (5,13) (4,14) (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L17:</td>
<td>(16,10) (15,10) (15,9) (15,8) (15,7) (15,6) (15,5) (15,4) (14,3) (13,3) (12,3) (11,3) (10,3) (9,3) (8,3) (7,3) (6,3) (5,3) (4,4) (4,5) (4,6) (4,7) (5,8) (5,9) (5,10) (5,11) (5,12) (6,12) (7,12) (8,12) (9,12) (10,12) (11,12) (11,13) (11,14) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L18:</td>
<td>(16,10) (16,11) (15,11) (15,12) (14,13) (14,14) (14,15) (13,15) (12,15) (11,14) (10,14) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L19:</td>
<td>(16,10) (16,9) (16,8) (15,7) (14,6) (13,5) (12,5) (11,4) (10,4) (9,5) (8,5) (7,6) (6,7) (6,8) (5,9) (5,10) (5,11) (6,12) (6,13) (7,14) (8,15) (9,16) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L20:</td>
<td>(16,10) (15,9) (15,8) (14,8) (13,8) (12,8) (11,8) (10,8) (9,9) (8,10) (8,11) (8,12) (8,13) (8,14) (9,14) (10,14) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L21:</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10) (9,11) (9,12) (9,13) (9,14) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L22:</td>
<td>(16,10) (16,11) (16,12) (15,13) (15,14) (14,14) (13,14) (12,15) (11,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L23:</td>
<td>(16,10) (15,9) (14,8) (13,7) (12,7) (11,6) (10,6) (9,6) (8,6) (7,7) (7,8) (7,9) (7,10) (7,11) (7,12) (7,13) (8,14) (9,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L24:</td>
<td>(16,10) (15,11) (14,12) (14,13) (13,14) (13,15) (12,15) (11,16) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L25:</td>
<td>(16,10) (16,9) (15,8) (14,7) (13,6) (12,6) (11,6) (10,6) (9,6) (8,7) (7,7) (6,8) (6,9) (5,10) (5,11) (5,12) (5,13) (6,14) (7,15) (8,16) (9,16) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L26:</td>
<td>(16,10) (15,9) (14,9) (13,8) (12,8) (11,8) (10,7) (9,7) (8,8) (7,9) (6,10) (6,11) (5,12) (5,13) (6,14) (7,15) (8,16) (9,16) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L27:</td>
<td>(16,10) (15,10) (14,10) (13,9) (12,9) (11,9) (10,9) (9,9) (8,9) (7,10) (7,11) (7,12) (6,13) (6,14) (7,15) (8,16) (9,16) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L28:</td>
<td>(16,10) (15,10) (14,10) (13,11) (13,12) (11,12) (11,13) (10,14) (10,15) (10,16) (12,12)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L29:</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (10,10) (10,9) (10,8) (10,7) (10,6) (10,5) (10,4) (10,3) (9,3) (8,3) (7,3) (6,3) (6,4) (6,5) (6,6) (6,7) (5,7) (4,7) (4,8) (4,9) (4,10) (4,11) (4,12) (4,13) (4,14) (4,15) (4,16) (5,15) (6,15) (7,15) (8,15) (9,15) (10,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L30:</td>
<td>(16,10) (16,9) (16,8) (15,7) (14,6) (13,5) (12,5) (11,4) (10,4) (9,4) (8,4) (7,5) (6,5) (5,6) (5,7) (5,8) (5,9) (6,10) (7,11) (8,11) (9,12) (10,12) (11,12) (9,13) (8,14) (9,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L31:</td>
<td>(16,10) (15,9) (14,9) (13,8) (12,8) (11,8) (10,7) (9,7) (8,7) (7,8) (6,9) (7,10) (8,11) (9,11) (9,12) (9,13) (9,14) (9,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L32:</td>
<td>(16,10) (16,9) (15,8) (14,7) (13,8) (13,9) (12,10) (11,10) (10,9) (9,8) (8,7) (7,8) (6,9) (5,10) (4,11) (5,12) (6,12) (7,14) (8,15) (9,16) (10,16) (7,13)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L33:</td>
<td>(16,10) (15,10) (14,11) (13,11) (12,12) (11,12) (10,12) (9,12) (8,13) (9,14) (9,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L34:</td>
<td>(16,10) (15,10) (14,10) (13,10) (12,10) (11,10) (11,11) (11,12) (11,13) (11,14) (11,15) (10,16)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
IBM 1800 MPX
1130 / 1800 BASIC FORTRAN IV
WITH SUBROUTINES RANDY, JOIN, AND PATOT

ACCEP TS NH H MATRICES AND NL L MATRICES
AND POSITIONS FOUR OF EACH TYPE, AFTER APPROPRIATE ROTATION,
IN A STANDARD 48 X 48 MATRIX, SO THAT A CONTINUOUS CLOSED FIGURE
IS FORMED

INTEGER HORL (8)
DIMENSION HATS (48, 48), MATH (50, 100), MATL (50, 100), IDA (3)

CALL RANDY (R)

READ DATE AND NUMBER OF PATTERNS
READ (2, 10 ) IDA, NOPAT
10 FORMAT (4I3)

READ PARAMETERS
NH IS THE NUMBER OF H MATRICES
NL IS THE NUMBER OF L MATRICES
READ (2, 20 ) NH, NL
20 FORMAT (2I5)

READ H AND L TYPE SUBMATRICES
READ (2, 30 ) ((MATH (I, J), J = 1, 100), I = 1, NH)
READ (2, 30 ) ((MATL (I, J), J = 1, 100), I = 1, NL)
30 FORMAT (25I3)

START LOOP OVER PATTERNS
DO 150 NO = 1, NOPAT

PRINT HEADING
WRITE (3, 120) NO, IDA
120 FORMAT (11H1VARGUS 10 A, 10x, 14H PATTERN NUMBER, I3, 10x, 3I3)

CALL JOIN (MATS, MATH, MATL, NH, NL, HORL)
WRITE(3,145)(HORL(IHX),IXX=1,8)
145 FORMAT ('0 L (1, 1) H (1, 2) L (1, 3) H (2, 3) L (3, 3) H (3, 1) L (3, 1) H (2, 1)' / 17, 7110)

C
C . PRINT PATTERN
CALL PATOT (MATS, 48, 48)
150 CONTINUE
C
CALL EXIT
END

VARIABLE ALLOCATIONS
R(R) = 0000 MATS(I) = 0901-0002 MATH(I) = 1C99-0902 MATL(I) = 3011-
NDP@T(I) = 301D NH(I) = 301E NL(I) = 301F I(I) = 3020
IXX(I) = 3023

STATEMENT ALLOCATIONS
10 = 302C 20 = 302F 30 = 3032 120 = 3035 145 = 304A 150 = 3107

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGEMS
IOCS

CALLED SUBPROGRAMS
RANDY JOIN PATOT MRED MRT MCOMP MIDA MIOX MIOI SUI

INTEGER CONSTANTS
2 = 3026 1 = 3027 100 = 3028 3 = 3029 8 = 302A 48 = 302B

CORE REQUIREMENTS FOR VAR10
COMMON 0 INSKEL COMMON 0
VARIABLES 12326 PROGRAM 236

END OF-compilation
VARGUS 10A

PATTERN NUMBER 1

L(1, 1) H(1, 2) L(1, 3) H(2, 3) L(3, 3) H(3, 2) L(3, 1) H(2, 1)
26 27 22 11 4 11 10 9

71
I. Identification

1. Subroutine Name: JØIN
2. Category: Pattern Generation
3. Purpose: Create pattern matrices by randomly selecting four corner submatrices and four side submatrices. These eight submatrices are joined together into a closed figure by performing the appropriate rotation and positioning and storing into the pattern matrix.
4. Date: August, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: See VARGUS 10A documentation.
4. Usage: CALL JØIN (MATS, MATH, MATL, NH, NL).
5. Subroutine Parameters:

   MATS: The 48 x 48 output matrix.
   MATH: The matrix each row of which is a string of subscript pairs for side submatrices.
   MATL: The matrix each row of which is a string of subscript pairs for corner submatrices.
   NH: The number of side submatrices.
   NL: The number of corner submatrices.
6. Output: \texttt{HRL}: A vector the eight elements of which are numbers corresponding to the corner and side submatrices chosen to form a particular pattern. The elements are in the order beginning at the upper left of the pattern and moving clockwise around the figure.

7. Subroutine Required:

\texttt{RANDY}: A pseudo-random number generator.

\texttt{ROTAT}: The subroutine used to rotate the side and corner submatrices so that they form a continuous closed figure.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.
SUBROUTINE JOIN (MATS, MATH, MATL, NH, NL, HORL)
INTEGER SUBMX (16, 16), CHANI (8), CHANJ (8), HORL (8)
DIMENSION MATS (48, 48), MATH (50, 100), MATL (50, 100)
DATA CHANI / 0, 0, 0, 16, 32, 32, 32, 16 /
DATA CHANJ / 0, 0, 16, 32, 32, 32, 16, 0 /
C
C CLEAR OUTPUT MATRIX MATS
DO 40 I = 1, 48
DO 40 J = 1, 48
40 MATS (I, J) = 0
C
DO 110 IBLOC = 1, 8
C
C CLEAR SUBMATRIX SUBMX
DO 50 I = 1, 16
DO 50 J = 1, 16
50 SUBMX (I, J) = 0
C
NTURN = (IBLOC - 1) / 2
CALL RANDY (R)
C
IF (IBLOC - 2 * (IBLOC / 2)) 60, 60, 80
C
60 I = R * NH + 1
DO 70 J = 1, 99, 2
K = MATH (I, J)
L = MATH (I, J + 1)
IF (K) 70, 100, 70
70 SUBMX (K, L) = 1
GO TO 100
C
80 I = R * NL + 1
DO 90 J = 1, 99, 2
K = MATL (I, J)
L = MATL (I, J + 1)
IF (K) 90, 100, 90
90 SUBMX (K, L) = 1
100 CALL ROTAT (SUBMX, 16, 16, NTURN)
HORL (IBLOC) = I
C
DO 110 I = 1, 16
DO 110 J = 1, 16
II = I + CHANI (IBLOC)
JJ = J + CHANJ (IBLOC)
110 MATS (II, JJ) = SUBMX (I, J)
RETURN
END
VARIABLE ALLOCATIONS
R(R )=0000  SUBMAX(I )=0103-0004 CHANI(I )=010B-0104 CHANJ(I )=0113-
IBLOC(I )=0116 NTURN(I )=0117 K(I )=0118 L(I )=0119

STATEMENT ALLOCATIONS
40 =0141 50 =016A 60 =01A1 70 =01CE 80 =01E5 90 =0212 100

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
RANDY ROTAT FADD FMPY FSTO IFIX FLOAT ISTOX STFAC SBF

INTEGER CONSTANTS
1=0120 43=0121 0=0122 8=0123 16=0124 2=0125

CORE REQUIREMENTS FOR JOIN
COMMON 0 INSKEL COMMON 0
VARIABLES 298 PROGRAM 354

END OF COMPILATION
I. Identification

1. Subroutine Name: ROTAT
2. Category: Pattern Modifier
3. Purpose: Accepts a matrix and rotates it by multiples of 90 degrees.
4. Date: July, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitation: Input matrix must be dimensioned to 16 x 16 in calling routine.
5. Subroutine Parameters:
   
   MX: The matrix to be rotated.
   NI: The number of rows.
   NJ: The number of columns.
   NTURN: The number of 90-degree rotations.

6. Subroutines Required:

   MODUL: performs modular arithmetic.

III. Mathematical Method and References

   Not Applicable.

IV. Listing of Source Program

   Attached.
V. Sample Run
   Attached.

VI. Remarks
   None.
MODUL

SUBROUTINE ROTAT (MX, NI, NJ, NTURN)
CALL MODUL (4, NTURN, NTURN)
NTURN = NTURN + 1
GO TO (120, 40, 60, 80), NTURN

90 DEGREE ROTATION
DO 50 I = 1, NJ
DO 50 J = 1, NI
K = NI - J + 1
50 LX (I, J) = MX (K, I)
GO TO 100

180 DEGREE ROTATION
DO 70 I = 1, NI
DO 70 J = 1, NJ
K = NI - I + 1
L = NJ - J + 1
70 LX (I, J) = MX (K, L)
M = NI
NI = NJ
NJ = M
GO TO 100

270 DEGREE ROTATION
DO 90 I = 1, NJ
DO 90 J = 1, NI
K = NJ - I + 1
90 LX (I, J) = MX (J, K)

LOAD ROTATED MATRIX
DO 110 J = 1, NI
MX (I, J) = LX (I, J)
110 CONTINUE
120 RETURN
END

VARIABLE ALLOCATIONS
LX(I) = 00FF-0000 I(I) = 0100 J(I) = 0101 K(I) = 0102

STATEMENT ALLOCATIONS
40 = 013D 50 = 014D 60 = 0175 70 = 018D 80 = 01C1 90 = 01D1 100

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
MODUL COMGO ISTDX SUBSC SUBIN

INTEGER CONSTANTS
4=0108 1=0109

CORE REQUIREMENTS FOR ROTAT
COMMON 0 INSKEL COMMON 0
VARIABLES 264 PROGRAM 278

END OF COMPILATION
I. Identification

1. Subroutine Name: MODUL
2. Category: Computational
3. Purpose: Perform modular arithmetic
4. Date: August, 1968
5. Programmer: Bob Brectenridge

II. Use Information

1. Language: 1309 Basic FORTRAN IV.
3. Limitations: Operates on one number per call.
4. Subroutine Usage: CALL 'MODUL (MODOP, IX, MODX).
5. Subroutine Parameters:
   MODOP: The modulus.
   IX: The input value.
   MODX: The output value.
6. Subroutines Required: None.

III. Mathematical Method and References

None.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.
VARIO
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13:707 HRS
*ONE WORD INTEGERS
*LIST ALL
*NONPROCESS PROGRAM
   SUBROUTINE MODUL (MODOP, IX, MODX)
   C SUBROUTINE FOR DOING MODULAR ARITHMETIC
   C MODOP IS THE MODULAR OPERATOR
   C IX IS THE INPUT VALUE
   C MODX IS THE OUTPUT MOD VALUE
   C
   MODX=IX
   GO TO 20
  10 MODX=MODX-MODOP
  20 IF (MODX-MODOP)30,10,10
  30 CONTINUE
   RETURN
END

STATEMENTS ALLOCATIONS
  10 =0013  20 =0019  30 =001F

FEATURES SUPPORTED
   NONPROCESS
   ONE WORD INTEGERS

CALLED SUBPROGRAMS
   SUBIN

CORE REQUIREMENTS FOR MODUL
   COMMON   O INSHEL COMMON   O
   VARIABLES   O PROGRAM   34

END OF COMPILATION
I. Identification

1. Subroutine Name: CØNG
2. Category: Comparison of Patterns
3. Purpose: Computes an estimate of the similarity of two VARGUS 10 patterns.
4. Date: August, 1969
5. Programmer: Earl M. Greer

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Matrices dimensioned to 48 x 48.
4. Subroutine Usage:
   CALL CØNG (MX, NI, NJ, LX, MI, MJ, IBLUR, CØNGR).
5. Subroutine Parameters:
   MX: Pattern matrix.
   NI: Number of rows.
   NJ: Number of columns.
   LX: Comparison matrix (typically the schema).
   MI: Number of rows in comparison matrix.
   MJ: Number of columns in comparison matrix.
   IBLUR: If IBLUR = 1, LX is blurred as well as MX.
   CØNGR: Returns the similarity measure.
6. Subroutines Required:
   BLUR: Blurs the input matrix using a 7 x 7 element random walk matrix.
III. Mathematical Method and References

$SUM$ contains a measure of the degree of overlap of the two patterns. $SUMZ$ contains a measure of the maximum possible overlap; i.e., a comparison of $LX$ with itself. $\%\text{NCR}$ is thus the percentage of maximum possible overlap of $LX$ with $LX$.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.
SUBROUTINE CONG (MX, NI, NJ, LX, MI, MJ, IBLUR, CONGR)

SUBROUTINE CONG

EARL M. GREER

AUGUST 1, 196

IBM 1000

ACCEPTS TWO MATRICES FROM CALLING STATEMENT,

BLURS THE FIRST, MULTIPLIES THE TWO MATRICES,

AND SUMS THE RESULTING MATRIX.

IF IBLUR = 1 THE SECOND MATRIX IS ALSO BLURRED ALONG WITH THE FIRST.

 THIS SECOND MATRIX IS SQUARED TO PROVIDE A STANDARD, THE MAXIMUM POSSIBLE CONGRUENCE.

THE DIFFERENCE BETWEEN THE STANDARD AND ACTUAL CONGRUENCE IS THE

DEVIAION, THE ACTUAL CONGRUENCE IS DIVIDED BY THE STANDARD AND MULTIPLIED BY 100 TO FIND THE PERCENT CONGRUENCE.

DIMENSION :LX(48,48), LX(48,48)

COMMON FX(54,54)

IF (IBLUR = 1) 60, 60, 60

50 CALL BLUR(LX, MI, MJ)

60 CALL BLUR(LX, NI, NJ)

SUM = 0.0

SUM2 = 0.0

DO 70 I = 1, NI

DO 70 J = 1, NJ

FX(I, J) = LX(I, J) * LX(I, J)

SUM = SUM + FX(I, J)

70 SUM = SUM + FX(I, J)

DO 90 I = 1, MI

DO 90 J = 1, MJ

FX(I, J) = LX(I, J) * LX(I, J)

SUM2 = SUM2 + FX(I, J)

90 SUM2 = SUM2 + FX(I, J)

DIFF = SUM2 - SUM

CONGR = (SUM / SUM2) * 100.0

RETURN

END

VARIABLE ALLOCATIONS

FX(RC) = FFFE-C938 SUM(R) = 0000 SUM2(R) = 0002 DIFF(R) = 0004

STATEMENT ALLOCATIONS

50 = 0030 60 = 0035 70 = 0062 90 = 0096

FEATURES SUPPORTED

NONPROCESS

ONE WORD INTEGERS

CALLED SUBPROGRAMS

BLUR FADDX FSUB FMPY FDIV FLD FSTD FSTOX FLOAT SU
REAL CONSTANTS
*000000E 00=003A  *100000E 03=000C

INTEGER CONSTANTS
1=000E

CORE REQUIREMENTS FOR CONG
COMMON 5832 INSKEI COMMON 0
VARIABLES 10 PROGRAM 188

END OF COMPILATION
VARGUS 10A PATTERN 2
VARGUS 10A PATTERN 3
RESULT OF CONG 0:1 VARGUS 10-A PATTERN 2 AND PATTERN 3 WITH IBLUR = 0
CONG RETURNS 322.63

RESULT OF CONG AS ABOVE WITH IBLUR = 1
CONG RETURNS 77.34
RESULT OF CONG ON VARGUS 10-A PATTERN 2 AND PATTERN 4 WITH IBLUR = 0
CONG RETURNS 353.69

RESULT OF CONG AS ABOVE WITH IBLUR = 1
CONG RETURNS 77.62
I. Identification

1. Subroutine Name: BLUR
2. Category: Pattern Manipulation
3. Purpose: To accept a binary matrix and simulate all possible 3-step random walks for every "black" cell. Thus a weight or loading is built up on a scale from 0 to 2 representing the probability that a given cell might be black.

4. Date: August, 1968
5. Programmer: Bob Brekenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Pattern matrix is a 48 x 48 fixed point matrix.


5. Subroutine Parameters:
   - MX: pattern matrix.
   - MI: number of rows.
   - NJ: number of columns.

6. Subroutines Required:
   - FSCAL: Rescales a matrix such that values are whole numbers from 0 to 2.
III. Mathematical Methods and References
None.

IV. Listing of Source Program
Attached.

V. Sample Run
Attached.

VI. Remarks
None.
RSCAL
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.580 HRS
*LIST ALL
*ONE WORD DEGENERATE
*NONPROCESS PROGRAM

SUBROUTINE BLUR(MX, NI, NJ)

C REAL IMX (54, 54)
DIMENSION *X (48, 48), IWALK (49)
COMMON IMX
DATA IWALK/1, 3, 6, 7, 6, 3, 3, 6, 10, 10, 6, 3, 6, 10, 25, 27, 25, 10, 6, 7, 10,
127, 30, 27, 10, 7, 6, 10, 25, 27, 25, 10, 6, 3, 6, 10, 10, 6, 3, 1, 3, 6, 7, 6, 3, 1/

C NJJ=NI+3
NII=NI+3
IN=NI+6
JN=NJ+6
C
C SET WORK MATRIX TO ZERO
DO 20 I=1, IN
DO 20 J=1, JN
20 IMX(I, J)=0
C
C CHECK INPUT MATRIX FOR SIGNAL
DO 50 I=1, 'II
DO 50 J=1, 'IJ
IF (MX(I, J) )50, 50, 30
30 IOWNER=I+6
JDOWN=J+6
KOUNT=1
DO 40 II=1, IOWNER
DO 40 JJ=JDOWN
IMX(II, JJ)=IMX(II, JJ)+IWALK(KOUNT)
40 KOUNT=KOUNT+1
50 CONTINUE
C
C SET BORDER AROUND MATRIX TO ZERO
DO 100 K=4, NJJ
IMX(4, K)=0
100 IMX(NII, K)=0
DO 105 K=4, NII
IMX(K, 4)=0
105 IMX(K, NJJ)=0
DO 110 K=4, NII
DO 110 J=4, NJJ
110 MX(K-3, J-3)=IMX(K, J)
CALL RSCAL (1, IMX, MX, NI, NJ)
RETURN
END

VARIABLE ALLOCATIONS
IMX(64)=FFFF-E938 IWALK(I )=0032-0002 NJJ(I )=0033 NII(I )=0034
I(I) = 0037  J(I) = 0038  IOVER(I) = 0039  JDOWN(I) = 003A
JJ(I) = 003D  K(I) = 003E

STATEMENT ALLOCATIONS
20 = 0078  30 = 00A9  40 = 00DE  50 = 00F6  100 = 0116  105 = 0139  11

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
RSCAL  FADD  FLDX  FSTO  FSTOX  IFIX  FLOAT  ISTOP  SUBSC  SUB

INTEGER CONSTANTS
3 = 0042  6 = 0043  1 = 0044  0 = 0045  4 = 0046

CORE REQUIREMENTS FOR BLUP
COMMON  S832  INSHEL COMMON  0
VARIABLES  66 PROGRAM  324

END OF COMPILATION
I. Identification

1. Subroutine Name: RSCAL
2. Category: Computational
3. Purpose: To rescale a matrix such that each cell contains a whole number in the interval 0 to 9.
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Matrix dimensions up to 48 x 48. Input matrix is either fixed or floating point. Output matrix is always fixed point.
5. Subroutine Parameters:
   
   MODE: Indicates whether data matrix is fixed point (MODE = 1), or floating point (MODE = 2).
   
   FMX: Floating-point input matrix.
   
   MX: Fixed-point input matrix. Returns rescaled matrix.
   
   NI: Number of rows.
   
   NJ: Number of columns.
6. Subroutines Required:

MIMAX: Find the maximum and minimum values in a matrix.

III. Mathematical Method and References

Simple scaling procedure.

IV. Listing of Source Program

Attached.

V. Sample Run

Attached.

VI. Remarks

None.
CON
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23,573 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM

SUBROUTINE RSCAL(MODE, FMX, MX, NI, NJ)
DIMENSION FMX(54, 54), MX(48, 48)
CALL MIMAX(MODE, FMX, MX, NI, NJ, BIG, SMALL)
DIVID=(BIG-SMALL)/9.9
DO 5 I=1, NI
DO 5 J=1, NJ
TEMP=(FMX(I, J)-SMALL)/DIVID
5 MX(I, J)=TEMP
RETURN
END

VARIABLE ALLOCATIONS
BIG(R )=0000 SMALL(R )=0002 DIVID(R )=0004 TEMP(R )=0006

STATEMENT ALLOCATIONS
5 =0048

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
MIMAX FSUB FDIV FLD FLDX FSTO IFIX ISTOX SUBSC SU

REAL CONSTANTS
.990000E 01=000C

INTEGER CONSTANTS
1=000E

CORE REQUIREMENTS FOR RSCAL
COMMON 0 INSKEL COMMON 0
VARIABLES 12 PROGRAM 92

END OF COMPILATION
I. Identification

1. Subroutine Name: MIMAX
2. Category: Computational
3. Purpose: Find the maximum and minimum values in a matrix.
4. Date: August, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Matrix dimensions up to 48 x 48.
5. Subroutine Parameters:
   
   **MODE:** Indicates whether data matrix is fixed point (MODE = 1), or floating point (MODE = 2).
   
   **FMX:** Floating-point input matrix.
   
   **MX:** Fixed-point input matrix.
   
   **NI:** Number of rows.
   
   **NJ:** Number of columns.
   
   **BIG:** Return largest value in matrix to calling program.
   
   **SMALL:** Returns smallest value in matrix to calling program.
6. Subroutines Required: None.

III. Mathematical Methods and References
   None.

IV. Listing of Source Program
   Attached.

V. Sample Run
   Attached.

VI. Remarks
   None.
BLUR
DMP FUNCTION COMPLETED
// FOR 01 JUL 71 23.590 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
SUBROUTINE MIMAX(MODE,FMX, Mx,NI,NJ,BIG,SMALL)
DIMENSION FMX(54,54),Mx(48,48)
GO TO (5,13),MODE
5 DO 7 I=1,NI
DO 7 J=1,NJ
7 FMX(I,J)=MX(I,J)
10 BIG=FMX(I,I)
SMALL=BIG
DO 15 I=1,NI
DO 15 J=1,NJ
IF(BIG-FMX(I,J)) 11,15,13
11 BIG=FMX(I,J)
GO TO 15
13 IF(SMALL-FMX(I,J)) 15,15,14
14 SMALL=FMX(I,J)
15 CONTINUE
RETURN
END

VARIABLE ALLOCATIONS
I(I )=0000
J(I )=0001

STATEMENT ALLOCATIONS
5 =0028 7 =0030 10 =0057 11 =0079 13 =0086 14 =0094 15

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
FSubX FLD FLDX FSTO FSTOX FLOAT COMGO LDFAc SUBSC SU

INTEGER CONSTANTS
1 =0004

CORE REQUIREMENTS FOR MIMAX
COMMON 0 INSKEl COMMON 0
VARIABLES 4 PROGRAM 176

END OF COMPILATION
I. Identification

1. Subroutine Name: SØLID
2. Category: Pattern Manipulation
3. Purpose: Accepts a closed figure represented in matrix form and fills in the interior.
4. Date: July, 1969
5. Programmer: David R. Harris

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: The program is designed to accept a 48 x 48 matrix. The size of IARY and JARY may be reduced to save space but at a cost in running time.
5. Subroutine Parameters:

   INARY: a 48 x 48 input matrix containing the pattern. This same matrix returns the solid figure so that the original pattern is destroyed.

6. Subroutines Required: None.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.
V. Sample Run
   Attached.

VI. Remarks
   None.
SUBROUTINE SOLID(INARY)
DIMENSION INARY(48,48), ARY(54,54), IARY(54), JARY(54)

C
DO 10 I=1,48
   DO 10 J=1,48
      ARY(I+2,J+2)= INARY(I,J)
10    THIS PUTS THE ORIGINAL ARRAY IN THE WORKING ARRAY
M1=52
M2=51
M3=50

C
DO 11 I=1,M1
   ARY(I,I)=-1
   ARY(M1,I)=-1
   ARY(I,M1)=-1
11    ARY(I,M1)=-1
C
THIS PUTS -1 AROUND THE BORDER
DO 15 I=2,M2
   ARY(2,I)=0
   ARY(M2,I)=0
   ARY(I,2)=0
15    ARY(I,M2)=0

C
IP1=1
IP2=2
IARY(1)=2
JARY(1)=2
ARY(2,2)=-1
C
THIS SECTION FILLS IN -1 AROUND THE FIGURE
35 K1=IARY(IP1)
   K2= JARY(IP1)
C
IF (ARY(K1+1,K2 )) 50,45,50
45 ARY(K1+1,K2 )=-1.
   IARY(IP2)=K1+1
   JARY(IP2)=K2
   IP2=IP2+1
50 IF(ARY(K1 ,K2-1)) 60,55,60
55 ARY(K1 ,K2-1)=-1.
   IARY(IP2)=K1
   JARY(IP2)=K2-1
   IP2=IP2+1
60 IF (ARY(K1 ,K2+1)) 70,65,70
65 ARY(K1 ,K2+1)=-1.
   IARY(IP2)=K1
   JARY(IP2)=K2+1
   IP2=IP2+1

103
70 IF (ARY(K1-1,K2)) 80,75,80
75 ARY(K1-1,K2)=-1
   IARY(IP2)=K1-1
   JARY(IP2)=K2
   IP2=IP2+1
80 IP1=IP1+1
   IF (IP1=IP2) 85,110,85
85 IF (IP2+4=K4) 95,90,90
90 IARY(IP2)=-1
   IP2=1
95 IF (IARY(IP1)) 100,100,35
100 IP1=1
   GO TO 35
110 DO 200 I=3,M3
   DO 200 J=3,M3
   IF (ARY(I,J)) 200,150,150
200 CONTINUE
   RETURN
END

VARIABLE ALLOCATIONS
ARY(I) = FFFE-C938 IARY(I) = 0035-0000 JARY(I) = 006B-0036 I(I) = 006C
M2(I) = 006F M3(I) = 0070 IP1(I) = 0071 IP2(I) = 0072

STATEMENT ALLOCATIONS
10 = 0094 11 = 00EE 15 = 0128 35 = 0158 45 = 0173 50 = 0196 55
75 = 0200 80 = 0223 85 = 022F 90 = 0237 95 = 0245 100 = 024E 110

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
FLD FLDX FSTOX FLOAT ISTOX LDFAc SUBSC SNR SUBIN

REAL CONSTANTS
.1000000E 01=007A

INTEGER CONSTANTS
1=007C 48=007D 52=007E 51=007F 50=0080 2=0081

CORE REQUIREMENTS FOR SOLID
COMMON 5832 INSKel COMMON 0
VARIABLES 122 PROGRAM 526

END OF COMPILATION
BASIC VARGUS 10A PATTERN
EFFECT OF SOLID ON PATTERN
I. Identification

1. Subroutine Name: WIDEN
2. Category: Pattern Manipulation
3. Purpose: To widen lines of a figure centered in a matrix.
4. Date: September, 1968
5. Programmer: Chip Bruce

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitation: Widens line one cell thick to 3 cells thick. Multiple calls to the subroutine will result in still thicker lines. Matrix should be dimensioned to 48 x 48.
5. Subroutine Parameters:

   MX: The matrix containing the pattern.
   NI: The number of rows.
   NJ: The number of columns.

6. Subroutines Required: None.

III. Mathematical Method and References

Each positive cell of the matrix is treated as the center of a 3 x 3 submatrix and its eight surrounding cells are set equal to 1.
IV. Listing of Source Program
   Attached.

V. Sample Run
   Attached.

VI. Remarks
   None.
SUBROUTINE WIDEN (MX, NI, NJ)

C ACCEPTS MATRIX MX (NI ROWS, NJ COLUMNS)
C AND PLACES 1 S IN EVERY POSITION OF A 3 X 3 SUBMATRIX OF MX
C WHICH IS CENTERED SUCCESSIVELY ON POSITIVE ENTRIES OF MX

DIMENSION MX (48, 48), LX (48, 48)
DO 5 I=1,NI
DO 5 J=1,NJ
5 LX(I,J)=0
DO 30 I = 1, NI
DO 30 J = 1, NJ
IF (MX (I, J)) 30s 30s
10 DO 20 K = 1, 3
DO 20 L = 1, 3
M = I + K - 2
N = J + L - 2
20 LX (M, N) = 1
30 CONTINUE
DO 40 I = 1, NI
DO 40 J = 1, NJ
MX (I, J) = LX (I, J)
40 CONTINUE
RETURN
END

VARIABLE ALLOCATIONS
LX(I )=08FF-0000  I(I )=0900  J(I )=0901  K(I )=0902
N(I )=0905

STATEMENT ALLOCATIONS
5 =0922 10 =0952 20 =096A 30 =0987 40 =09AC

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
ISTOX SUBSC SUBIN

INTEGER CONSTANTS
1=0908  0=0909  3=090A  2=090B

CORE REQUIREMENTS FOR WIDEN
COMMON 0 INSKEL COMMON 0
VARIABLES 2312 PROGRAM 184
EFFECT OF WIDEN ON PATTERN
I. Identification

1. Subroutine Name: WIDER
2. Category: Pattern Manipulation
3. Purpose: To widen lines of a closed figure inwardly and retain the original figure as the outer perimeter.
4. Date: September, 1968
5. Programmer: Bob Breckenridge

II. Use Information

1. Language: 1800 Basic FORTRAN IV.
3. Limitations: Widens to 3 cells thickness. Matrix should be dimensioned to 48 x 48.
5. Subroutine Parameters:
   
   MX: pattern matrix.
   NI: number of rows.
   NJ: number of columns.
6. Subroutines Required:
   
   SOLID: Fills in closed figure.
   WIDEN: Thickens lines to width 3.

III. Mathematical Methods and References

None.

IV. Listing of Source Program

Attached.
V. Sample Run
   Attached.

VI. Remarks
   None.
WIDEN
DMP FUNCTION COMPLETED
// FOR WIDER 30 JUN 71 13.887 HRS
*NONPROCESS PROGRAM
*LIST ALL
*ONE WORD INTEGERS

SUBROUTINE WIDER (MX,NI,NJ)
DIMENSION MX(48,48),LX(48,48)
DO 5 I=1,NI
DO 5 J=1,NJ
5 LX(I,J)=MX(I,J)
CALL SOLID(LX)
CALL WIDEN(MX,NI,NJ)
CALL WIDEN(MX,NI,NJ)
DO 6 I=1,NI
DO 6 J=1,NJ
6 MX(I,J)=MX(I,J)*LX(I,J)
RETURN
END

VARIABLE ALLOCATIONS
LX(I )=08FF-0000 I(I )=0900 J(I )=0901

STATEMENT ALLOCATIONS
5 =0920 6 =0952

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
SOLID WIDEN ISTOX SUBSC SUBIN

INTEGER CONSTANTS
1=0904

CORE REQUIREMENTS FOR WIDER
COMMON 0 INSKEL COMMON 0
VARIABLES 2308 PROGRAM 112

END OF COMPILATION
BASIC VARGUS 10A PATTERN
EFFECT OF WIDER ON PATTERN
I. Identification

1. Subroutine Name: RANDY
2. Category: Pseudo-random Number Generator
3. Purpose: Compute uniformly distributed random floating-point numbers between 0.0 and 1.0.
4. Date: March 25, 1968
5. Programmer: Bill Randol Brown
   Statistical tests in Remarks Section performed by Mike Abbamonte, April 20, 1971.

II. Use Information

1. Language: IBM Basic Fortran IV.
3. Limitations: None.
4. Subroutine Usage: CALL RANDY (R).
5. Input Card: START card.
   a. The START card contains 68 random digits in columns 1 to 68. Within these 68 digits strings of zeros in excess of two should not occur.
   b. In columns 69-72 of the card should be four digits sampled at random and with replacement from 1, 3, 7 and 9.
   c. This START card is read in by the subroutine itself. This read occurs only the first time
the subroutine is called. It is recommended that RANDY be called once early in execution so that the READ statement will be executed at a predictable time and thereby data synchronization problems avoided.

6. Output: The value returned (R) will be a floating-point number between 0.0 and 1.0.

7. Subroutines required: None.

III. Mathematical Methods and References

a. Seventeen generators each operating under the power residue method are in RANDY. The random number generated on call i is used to select which of the seventeen generators will be used on call i+1.

b. The cycle of each generator is 8192.


IV. Listing of Source Program

Attached.

V. Sample Output

Attached.

VI. Remarks

Using four randomly selected START cards, four Chi square tests were performed, each involving the
generation of 20,000 random numbers and 128 degrees of freedom

Test #1. Chi square/df = .900 ; 22nd percentile
Test #2. Chi square/df = .725 ; 1st percentile
Test #3. Chi square/df = .937 ; 30th percentile
Test #4. Chi square/df = .917 ; 37th percentile

It should be noted that these numbers tend to fall on the low side of the Chi square distribution indicating that the numbers generated by RANDY tend to be more evenly distributed in the unit interval that you would expect when randomly sampling from a rectangular distribution.

RANDY was also tested for cycling and none was found in the first 100,000 numbers. The criterion for cycling was repetition of the second through the eleventh numbers.
RAN
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13.478 HRS
*NONPROCESS PROGRAM
*ONE WORD INTEGERS
*LIST ALL

SUBROUTINE RANDY(RR)
DIMENSION KU(17), KC(17), KL(4)
DATA NDUM, KC /1, 899, 835, 957, 877, 915, 861, 947, 925, 867, 909, 883,
1851, 941, 893, 931, 845, 963/
GO TO (15, 16), NDUM
15 READ(2, 19) (KU(I), I=1, 17), (KL(J), J=1, 4)
19 FORMAT (17I4, 4I1)
J=0
DO 13 I=1, 17
J=J+1
KU(I)=KU(I)*10+KL(J)
IF(KU(I))21, 22, 22
21 KU(I)=KU(I)+32767+1
22 IF(J-4)13, 11, 11
11 J=0
13 CONTINUE
R=KU(1)
R=R/32768.
NDUM=NDUM+1
16 K=R/16.99+1.0
IR=KU(K)*KC(K)
IF(IR)23, 24, 24
23 IR=IR+32767+1
24 R=IR
R=R/32768.
KU(K)=IR
RR=R
RETURN
END

VARIABLE ALLOCATIONS
R(R )=0000  KU(I )=0012-0002  KC(I )=0023-0013  KL(I )=0027
J(I )=002A  K(I )=0028  IR(I )=002C

STATEMENT ALLOCATIONS
19 =003D  15 =004D  21 =00AD  22 =00AD  11 =00B3  13 =00B7  16

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CALLED SUBPROGRAMS
FADD FMUL FDIV FLD FSTO IFIX FLOAT COMGO ISTOX MR1

REAL CONSTANTS
*327680E 05=0030  *169900E 02=0032  *100000E 01=0034

INTEGER CONSTANTS
CORE REQUIREMENTS FOR RANDY
COMMON 0 INSKEL COMMON 0
VARIABLES 48 PROGRAM 222

END OF COMPILATION
VALUES RETURNED FROM RANDY

0.46206670
0.97213757
0.60293591
0.33059698
0.09817506
0.82565319
0.71530163
0.06094361
0.42031866
0.22714236
0.80990612
0.48190313
0.71408092
0.34793096
0.52871716
0.10806275
0.96603405
0.62716686
0.91708385
0.30319219
0.22537234
0.28756719
0.87167370
0.42570934
0.96157848
0.99996960
0.97061169
0.69894421
0.26358038
0.58132946
0.74215710
0.40298467
0.22409060
0.19638064
0.22579959
0.02621460
0.07809449
0.63833630
0.78494274
0.33944708
I. Identification

1. Subroutine Name: PAT@T
2. Category: Pattern Construction
3. Purpose: Output a 48 x 48 matrix in two symbols.
4. Date: July, 1968

II. Use Information

1. Language: 1800 Basic FØRTRAN IV.
3. Limitations: Accepts fixed point 48 x 48 matrix.
   Positive numbers are output by over-printing a zero by an asterisk. Zero
   or negative numbers are output as a blank space.
4. Subroutine Usage: CALL PAT@T (MX, NI, NJ)
5. Subroutine Parameters:
   MX: Matrix to be output. Must be dimensioned to 48 x 48.
   NI: Number of rows to output.
   NJ: Number of columns to output.
6. Subroutines Required: None.

III. Mathematical Methods and References

Not Applicable.

IV. Listing of Source Program

Attached.
V. Sample Run
   Attached.

VI. Remarks
   None.
JOIN
DMP FUNCTION COMPLETED
// FOR 30 JUN 71 13.732 HRS
*LIST ALL
*ONE WORD INTEGERS
*NONPROCESS PROGRAM

SUBROUTINE PATOT(MX,N1,NJ)
C ACCEPTS MATRIX MX(NI ROWS, NJ COLUMNS)
C AND PRINTS ALL POSITIVE ENTRIES AS BLACK SPOTS,
C BY OVER PRINTING CHA AND CHB FOR BLACK AND CHC FOR WHITE.
DIMENSION :IX(48,48), T(48), S(48)
DATA CHA,C1B,CHC/'0','*',' ','/'
DO 100 I=1, N1
DO 50 J=1, NJ
IF(MX(I,J))30, 30, 40
30 T(J)=CHC
S(J)=CHC
GO TO 50
40 T(J)=CHA
S(J)=CHB
50 CONTINUE
WRITE (3,51)(T(J), J=1, NJ)
51 FORMAT (1H 48A1)
WRITE (3,52)(S(J), J=1, NJ)
52 FORMAT (1H+48A1)
100 CONTINUE
RETURN
END

VARIABLE ALLOCATIONS
T(R )=005E-0000 S(R )=00RE-0060 CHC(R )=00C0 CHA(R )=00C2
J(I )=00C7

STATEMENT ALLOCATIONS
30 =00CC 50 =00D1 30 =00F4 40 =0105 50 =0114 100 =014B

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
CALLED SUBPROGRAMS
FLD FSTUX MWRT MCOMP MIOFX SUBSC SUBIN

INTEGER CONSTANTS
1=00CA 3=00CB

CORE REQUIREMENTS FOR PATOT
COMMON 0 INSKEL COMMON 0
VARIABLES 202 PROGRAM 140

END OF COMPIIATION
DISTRIBUTION LIST

CG, USAMC, Wash, D. C.
AMCDL (Ofc of Dep for Labs)
AMCRD (Air Def & Msl Ofs)
AMCRD (Mobility Ofs)
AMCRD (Comm-Elec Ofs)
AMCRD-G
AMCRD (Weapons Ofs)
AMCRD (Dr. Kaufman)
AMCRD (Mr. Crellin)

Ofc of Chief of Staff, DA, Wash, D. C.
CSAVCS-W-TIS

USA Behavioral Science Rsch Lab
Arlington, Va.

Dr. J. E. Uhlaner, Dir.
USA Behavioral Science Rsch Lab
Arlington, Va.

Behavioral Sciences Division
Ofc, Chief of R&D, DA
Washington, D. C.

Deputy Chief of Staff for Personnel
DA, Washington, D. C.
Personnel Rsch Div.

CG, USACDC, Fort Belvoir, Va.
CDCDC-CD
CDCMR
CDCRE

CO, USACDC Air Defense Agency
Fort Belvoir, Texas

CO, USACDC Artillery Agency
Fort Sill, Okla.

CO, USACDC Aviation Agency
Fort Rucker, Alabama

CO, USACDC CBR Agency
Fort McClellan, Alabama

CG, USACDC Combat Arms Group
Fort Leavenworth, Kansas

CG, USACDC Combat Svc Spt Group
Fort Lee, Va.

CG, USACDC Comm-Elec Agency
Fort Monmouth, N. J.

CO, USACDC Engineer Agency
Fort Belvoir, Va.

CO, USACDC
Institute of Strategic & Stability Ops
Fort Bragg, N. C.

CO, USACDC Med Svc Agency
Fort Sam Houston, Texas

CO, USACDC Military Police Agency
Fort Gordon, Georgia

CO, USACDC Supply Agency
Fort Lee, Va.

USACDC Experimentation Command
Tech Library, Box 22

CO, USA Environ Hygiene Agency
Edgewood Arsenal, Md.

CO, USA Edgewood Arsenal, Md.

CO, Picatinny Ars, Dover, N. J.

CO, Frankford Arsenal, Phila, Pa.

CO, Watervliet Arsenal, N. Y.

CO, Waterid For R&D Lab
Fort Totten, Flushing, L. I., N. Y.

CO, USA Med Equip R&D Lab
Fort Totten, Flushing, L. I., N. Y.

CO, USA Med Rsch Lab
Fort Knox, Kentucky

CO, Harry Diamond Labs
Washington, D. C.

CO, USA Mobility Equip R&D Ctr
Fort Belvoir, Va.

CO, USA Mobility Equip R&D Ctr
Fort Belvoir, Va.

CO, USA Mobility Equip R&D Ctr
Fort Belvoir, Va.

CO, USA Mobility Equip R&D Ctr
Fort Belvoir, Va.

CO, USA Mobility Equip R&D Ctr
Fort Belvoir, Va.

U. S. Army Natick Laboratories
Natick, Mass.

USA Gen Equip Test Activity

USA Bd for Avn Acct Rsch Lab
Fort Rucker, Ala.

USA Bd for Avn Acct Rsch Lab
Fort Rucker, Ala.

USA Gen Equip Test Activity

USA Gen Equip Test Activity

President, USA Infantry Board
Fort Benning, Georgia

President, USA Infantry Board
Fort Benning, Georgia

President, USA Maintenance Bd.
Fort Knox, Kentucky

President, USA Maintenance Bd.
Fort Knox, Kentucky

USA Armor, HRU, Fort Knox, Ky.

CO, USA Med Rsch Lab
Fort Knox, Kentucky
<table>
<thead>
<tr>
<th>Institution</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>CG, USA Weapons Command, RI, Ill.</td>
<td>AMSWE-RDT</td>
</tr>
<tr>
<td></td>
<td>AMSWE-SMM-P</td>
</tr>
<tr>
<td></td>
<td>SWERI-RDD-PD</td>
</tr>
<tr>
<td>CG, USA Tank-Automotive Command, Warren, Michigan</td>
<td>AMSTA-R</td>
</tr>
<tr>
<td></td>
<td>AMSTA-RHFL</td>
</tr>
<tr>
<td></td>
<td>AMSTA-RKAE</td>
</tr>
<tr>
<td>Cleveland Army Tank-Auto Plant, Cleveland, Ohio</td>
<td>HF Engr.</td>
</tr>
<tr>
<td>Director of Research</td>
<td>HumRRO Div. No. 5 (Air Defense)</td>
</tr>
<tr>
<td>PO Box 6021, Fort Bliss, Texas</td>
<td></td>
</tr>
<tr>
<td>Comdt., USA Artillery &amp; Msl School, Fort Sill, Oklahoma</td>
<td>USAAMS Tech Library</td>
</tr>
<tr>
<td>Tech Ref Div</td>
<td></td>
</tr>
<tr>
<td>Tech Ref Div</td>
<td></td>
</tr>
<tr>
<td>CO, Yuma Proving Ground, Yuma, Ariz.</td>
<td>Technical Library</td>
</tr>
<tr>
<td>CO, USA Tropic Test Center, PO Drwr 942, Fort Clayton, CZ</td>
<td>Behavioral Scientist</td>
</tr>
<tr>
<td>CO, USA Arctic Test Center, APO Seattle, Wash.</td>
<td>STEAC-IT</td>
</tr>
<tr>
<td>USA Standardization Group, UK</td>
<td>Box 65, FPO New York</td>
</tr>
<tr>
<td>USATECOM, Bldg 314, APG</td>
<td>USACDC Liaison Ofc, Bldg 314, APG</td>
</tr>
<tr>
<td>CO, USACDCMA, Bldg 305, APG</td>
<td>US Marine Liaison Ofc, Bldg 314</td>
</tr>
<tr>
<td>Tech Library, Bldg 305, APG</td>
<td>Dir, Naval Research Laboratory, Washington, D.C.</td>
</tr>
<tr>
<td></td>
<td>Code 5143A</td>
</tr>
<tr>
<td></td>
<td>Code 455 Ofc of Naval Research, Washington, D.C.</td>
</tr>
<tr>
<td></td>
<td>Engr Psychol (Dr. Tolcott)</td>
</tr>
<tr>
<td></td>
<td>Code 458 Ofc of Naval Research, Washington, D.C.</td>
</tr>
<tr>
<td></td>
<td>Personnel &amp; Tng (Dr. Farr)</td>
</tr>
<tr>
<td></td>
<td>USN Submarine Med Ctr, Libr Box 600, USN Sub Base, Groton, Conn.</td>
</tr>
<tr>
<td></td>
<td>CO &amp; Dir, Naval Tng Dev Ctr, Orlando, Florida</td>
</tr>
<tr>
<td></td>
<td>Technical Library</td>
</tr>
<tr>
<td></td>
<td>US Navy Electronics Laboratory, San Diego, Calif.</td>
</tr>
<tr>
<td></td>
<td>Ch, Human Factors Div.</td>
</tr>
<tr>
<td></td>
<td>Code 3400 (Wardell B. Welch)</td>
</tr>
<tr>
<td></td>
<td>Dept of Operations Analysis, Naval Postgraduate School, Monterey, Calif.</td>
</tr>
<tr>
<td></td>
<td>Prof. James K. Arima</td>
</tr>
<tr>
<td></td>
<td>RADC (EMEDI), Griffiss AFB, N.Y.</td>
</tr>
<tr>
<td></td>
<td>Hq, ESD (ESTI), L. G. Hanscom Field, Bedford, Mass.</td>
</tr>
<tr>
<td></td>
<td>Wright-Patterson AFB, Ohio</td>
</tr>
<tr>
<td></td>
<td>6570 AMRL (MRHE)</td>
</tr>
<tr>
<td></td>
<td>6570 AMRL (MRHER/Bates)</td>
</tr>
<tr>
<td></td>
<td>6570 AMRL (MRHE/Warrick)</td>
</tr>
<tr>
<td></td>
<td>AFFDL-FDCR (CDIC)</td>
</tr>
<tr>
<td></td>
<td>AMD (AMRH) Brooks AFB, Texas</td>
</tr>
<tr>
<td></td>
<td>Hq, 4442D Combat Crew Tng Wing (TAC) Little Rock AFB, Jacksonville, Ark.</td>
</tr>
<tr>
<td></td>
<td>Central Files</td>
</tr>
<tr>
<td></td>
<td>Civil Aeromedical Institute, Fed Avn Agency Aero Center, PO Box 25082, Okla City, Okla.</td>
</tr>
<tr>
<td></td>
<td>Psychol Br, AC-118</td>
</tr>
<tr>
<td></td>
<td>Rach Sec, Psychology Service, VA Hospital, Irving Ave &amp; Univ Pl.</td>
</tr>
<tr>
<td></td>
<td>Syracuse, N.Y.</td>
</tr>
<tr>
<td></td>
<td>Dr. Harvey A. Taub</td>
</tr>
<tr>
<td></td>
<td>Federal Aviation Administration, 800 Independence Ave, S.W. Washington, D.C.</td>
</tr>
<tr>
<td></td>
<td>Admin Stds Div (MS-110)</td>
</tr>
<tr>
<td></td>
<td>USPO Dept, Bureau R&amp;E, HF Br. Washington, D.C.</td>
</tr>
<tr>
<td></td>
<td>Mr. D. Cornog</td>
</tr>
<tr>
<td></td>
<td>Defense Documentation Center, Cameron Station, Alexandria, Va.</td>
</tr>
<tr>
<td></td>
<td>American Institute for Rsch, 8555 16th St, Silver Spring, Md.</td>
</tr>
<tr>
<td></td>
<td>Library</td>
</tr>
<tr>
<td></td>
<td>American Institute for Rsch, 135 North Bellefield, Pgh, Pa.</td>
</tr>
<tr>
<td></td>
<td>Library</td>
</tr>
<tr>
<td></td>
<td>American Institute for Rsch, PO Box 1113, Palo Alto, Calif.</td>
</tr>
<tr>
<td></td>
<td>Library</td>
</tr>
<tr>
<td></td>
<td>Center for Rsch in Social Systems</td>
</tr>
<tr>
<td></td>
<td>American Institutes for Rsch, 10605 Concord St, Kensington, Md.</td>
</tr>
<tr>
<td></td>
<td>ISB</td>
</tr>
<tr>
<td></td>
<td>The Franklin Institute Rsch Labs, Phila, Pa.</td>
</tr>
<tr>
<td></td>
<td>Tech Reports Library</td>
</tr>
<tr>
<td></td>
<td>Dr. J. Oransky</td>
</tr>
<tr>
<td></td>
<td>Serials Unit, Purdue University Lafayette, Ind.</td>
</tr>
<tr>
<td></td>
<td>Dept Psychol, Univ of Maryland College Park, Md.</td>
</tr>
<tr>
<td></td>
<td>Mr. R. K. Brome, Govt Pub Sec JFK Memorial Library</td>
</tr>
<tr>
<td></td>
<td>Calif State College/Los Angeles, Los Angeles, Calif.</td>
</tr>
<tr>
<td></td>
<td>Dr. R. C. Pearson</td>
</tr>
<tr>
<td></td>
<td>Dept of Ind Engineering, North Carolina State Univ.</td>
</tr>
<tr>
<td></td>
<td>Raleigh, N.C.</td>
</tr>
<tr>
<td>Name</td>
<td>Title/Role</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dr. F. Loren Smith</td>
<td>Dept Psychol, Univ of Delaware</td>
</tr>
<tr>
<td>Dr. H. W. Stoudt</td>
<td>Harvard Univ, Boston, Mass.</td>
</tr>
<tr>
<td>Dr. Leonard Uhr</td>
<td>Computer Sciences Dept, Univ of Wisconsin</td>
</tr>
<tr>
<td>Dr. R. A. Wunderlich</td>
<td>Psychol Dept, Catholic Univ, Washington, D.C.</td>
</tr>
<tr>
<td>Psychological Abstracts</td>
<td></td>
</tr>
<tr>
<td>AC Electronics Div, GMC</td>
<td></td>
</tr>
<tr>
<td>J. S. Inserra, HF</td>
<td>Tech Library, Dept 32-55 2A</td>
</tr>
<tr>
<td>Libr, Chrysler Def Engr</td>
<td></td>
</tr>
<tr>
<td>Grumman Aircraft Engr Corp</td>
<td></td>
</tr>
<tr>
<td>Hughes Aircraft Co.</td>
<td></td>
</tr>
<tr>
<td>Hughes Aircraft Co.</td>
<td></td>
</tr>
<tr>
<td>Hughes Aircraft Co.</td>
<td></td>
</tr>
<tr>
<td>Dr. Lauritz S. Larsen</td>
<td>Univ of SC, College of Engr, Traffic &amp; Transportation Ctr, Columbia, S.C.</td>
</tr>
<tr>
<td>Dr. Irwin Pollack</td>
<td>Univ of Mich, Ann Arbor, Mich.</td>
</tr>
<tr>
<td>Doc Libr, Wilson Library</td>
<td></td>
</tr>
<tr>
<td>Rsc Analysis Corp, McLean, Va.</td>
<td>Document Library</td>
</tr>
<tr>
<td>Rsc Analysis Corp, McLean, Va.</td>
<td>Document Library</td>
</tr>
<tr>
<td>Ritchie, Inc, Dayton, Ohio</td>
<td></td>
</tr>
<tr>
<td>Dr. Herbert J. Bauer</td>
<td>GM Rsch Labs, GM Tech Ctr Warrent, Mich.</td>
</tr>
<tr>
<td>Dr. Edwin Cohen</td>
<td>Link Group, Gen Precision Sys Inc, Binghamton, New York</td>
</tr>
<tr>
<td>Mr. Henry E. Gutmann</td>
<td>Sandia Corp, Albuquerque, N.M.</td>
</tr>
<tr>
<td>Dr. M. I. Kurke</td>
<td>Human Sciences Rsch Inc, McLean, Va.</td>
</tr>
<tr>
<td>Mr. James Moreland</td>
<td>Westinghouse Elec Corp, R&amp;D Ctr, Pittsburgh, Pa.</td>
</tr>
<tr>
<td>Mr. Robert F. Roser</td>
<td>Resors Rsch Association, Upland, Calif.</td>
</tr>
<tr>
<td>Dr. S. Seidenstein, Org 55-60</td>
<td>Bldg 151, Lockheed, PO Box 504 Sunnyvale, Calif.</td>
</tr>
<tr>
<td>Mr. Wesley E. Woodson</td>
<td>MAN Factors, Inc, San Diego, Calif.</td>
</tr>
<tr>
<td>Mr. C. E. Righter</td>
<td>AIresearch Mfg Co, Los Angeles, Calif.</td>
</tr>
<tr>
<td>Dr. Charles Abrams</td>
<td>Human Factors Rach, Goleta, Calif.</td>
</tr>
<tr>
<td>Dr. Corwin A. Bennett</td>
<td>Kansas State University, Manhattan, Kansas</td>
</tr>
<tr>
<td>The University of Wyoming</td>
<td>Documents Library, Laramie, Wyoming</td>
</tr>
<tr>
<td>Dr. Lawrence C. Perlmutter</td>
<td>Bowdoin College, Brunswick, Maine</td>
</tr>
<tr>
<td>Dr. Alexis M. Anikeeff</td>
<td>Univ of Akron, Akron, Ohio</td>
</tr>
<tr>
<td>The Boeing Co, Vertol Division</td>
<td>Document Library</td>
</tr>
<tr>
<td>Mr. Walter Jablonski</td>
<td>Grumman Aerospace Corp, Bethpage, N.Y.</td>
</tr>
<tr>
<td>Mr. Gerald J. Fox</td>
<td>BioTechnology, Inc.</td>
</tr>
<tr>
<td>Dr. Bill R. Brown</td>
<td>University of Louisville, Louisville, Kentucky</td>
</tr>
<tr>
<td>Mr. John H. Duddy, Dept 62-40</td>
<td>Bldg 151, Lockheed, PO Box 504 Sunnyvale, Calif.</td>
</tr>
<tr>
<td>Dr. Arthur S. Kamlet</td>
<td>Bell Telephone Labs (IB-125), Whippany, N.J.</td>
</tr>
<tr>
<td>US Dept of Commerce</td>
<td></td>
</tr>
<tr>
<td>Dr. Arthur Rubin</td>
<td></td>
</tr>
<tr>
<td>Dr. Corwin A. Bennett</td>
<td>Kansas State University, Manhattan, Kansas</td>
</tr>
<tr>
<td>The University of Wyoming</td>
<td>Documents Library, Laramie, Wyoming</td>
</tr>
<tr>
<td>Dr. Lawrence C. Perlmutter</td>
<td>Bowdoin College, Brunswick, Maine</td>
</tr>
<tr>
<td>Dr. Alexis M. Anikeeff</td>
<td>Univ of Akron, Akron, Ohio</td>
</tr>
<tr>
<td>The Boeing Co, Vertol Division</td>
<td>Document Library</td>
</tr>
<tr>
<td>Mr. Walter Jablonski</td>
<td>Grumman Aerospace Corp, Bethpage, N.Y.</td>
</tr>
<tr>
<td>Mr. Gerald J. Fox</td>
<td>BioTechnology, Inc.</td>
</tr>
<tr>
<td>Dr. Arthur Rubin</td>
<td></td>
</tr>
</tbody>
</table>
This document describes a collection of computer programs developed for use in research on human pattern perception. The overall orientation which guided the development of the VARGUS (Variable Generator of Unfamiliar Stimuli) pattern-generation programs and the historical backgrounds of each category (VARGUS 7, 9 and 10) are related in the first section. The second section provides documentation, sample output and summary for each program and subroutine.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pattern Recognition</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Photointerpretation</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Image Interpretation</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Man-Machine Decision Making</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Feature Space</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Feature Selection</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Personnel Selection</td>
<td>8</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>