デザインのための3種のアルゴリズムについて

鹿児島大学理学部紀要（数学・物理学・化学），
No. 9, p. 19–29, 1976.

デザインのための3種のアルゴリズムについて

富樫　昭*・藤野精一**

（1976年9月28日受理）

On Three Algorithms for Design by Computer

Akira Togasi and Seiiti Huzino

Abstract

In this paper we shall propose three methods of automatically designing by computer, and, by programming these methods, we shall investigate their advantages and shortcomings.

It is shown that the method of using p-ary number system (so called algorithm (C)) is superior than others.

Examples of programs by FORTRAN and results are given.

1°. はじめに

コンピュータを使用して情報処理を実行するとき，でてくる結果があらかじめ予想がつく場合と，どのような結果が生ずるか未知の場合がある。前者には，出力形式を定めて文字や数字を印刷したり，記号を表示するようなことがしばしば生ずる事務計算の場合が考えられ，後者の代表的なものとしては数値計算の問題の大部分が該当する。いずれにしても，コンピュータを使用する利点は，当面の情報処理を人間が実行するよりはるかに高速度に，かつ，はるかに正確に実行できる点である。とくに，単調な仕事を反復的に実行しなければならないような場合には，コンピュータはその偉力を最大限に発揮する。標記の“デザインをコンピュータで画かせる問題”は，数値計算の場合と異なり，コンピュータの非数値的な応用の一つであり，最初からどんな図形を画かせようかという，目標のわかった情報処理である。一見，比較的単純な問題のようにみえるが，決してそうではない。

この論文の目的は，デザイン，とくに各種の模様をコンピュータで自由に画かせるための3種のアルゴリズムを開発し，考えられたアルゴリズムを，実際にプログラミングしてその優劣を比較し，この方面の問題の探究のための足がかりを作り，今後の参考とすることである。

2°. アルゴリズム作成の方針

出力用紙一頁分全体に模様，たとえば，市松模様，山形模様，亀甲模様のような昔からよく知られた模様を画かせるにはどうしたらよいか？　問題の出発点をここにおこう。考えられる作成法としてまず考えられるのは，基本となる図形の最小単位を決定し，これを，たて，よこに反復するということである。上の模様の場合は，比較的容易にこれら最小基本単位が求めら

* 鹿児島大学理学部数学教室 (Department of Mathematics, Kagoshima University)
** 九州大学理学部数学教室 (Department of Mathematics, Kyushu University)

れる。もうすこし複雑な図形の場合は，基本単位の決定が困難となるであろう。しかし，これらの基本単位を何等かの方法で数種用意すれば，基本単位の組合わせ如何により，種々の模様が混合された，かなり面白い結果が得られるであろう。したがって，単純な発想ではあるが，これは考察に値する1つのアルゴリズムを構成する。これをアルゴリズム (A) とする。

アルゴリズム (A) の方針は

"各種模様に応じてそれに対する最小基本単位を決定し，それらを構成するサブルーチンを作成し，その組合わせを種々変更差し替えをして新しい模様のサブルーチンを合成する" ことである。

この方法 (A) では各種模様に応じたサブルーチンを多数作成しておくことが必要であり，この作成にかなり労力を必要とする。そこで，むしろ模様によって変更しない一つの主ルーチンを作り，これの利用により，データとして，実際の模様の切片を挿入して全模様を印刷しようという動きが生ずるのは自然な成り行きであろう。この方法をアルゴリズム (B) と名づける。

アルゴリズム (B) の方針は

"模様の基本単位を文字形式の入力として，データカードより入力して，全模様を印刷する" ことである。

アルゴリズム (A) とことなり，(B) では，任意の模様を作成するのに一段と便利となっている。この2つの方法の長短を比較することにより，次のような最終的と思われる案に到達した。それは結局，各行ごとに記号を印刷する能率のよい方法を案出すればよいという点に注目して，各行を一定長の有限個のブロックに分割し，各ブロックを構成する模様を記号列とみなすことから出発する。これら記号列の各記号に数値，0, 1, $\cdots\cdots$ p-1 (p は模様を構成する記号の総数) を付与することにすると，これらのブロックに p 進数が1意に対応することになる。したがって，この p 進数をデータとして入れて，p 進数値の各桁に，もとの対応する記号を印刷することにすれば，有限個の p 進数の列を指定することによって，任意の模様を印刷できることになる。

実際には入力としてはこの p 進数に対応する10進数を入れ，内部でこの10進数を対応する p 進数に変換して使用すれば，p 進数に不慣れな人にとっても何等使用上の不便は生じないであろう。これをアルゴリズム (C) と名づけよう。

アルゴリズム (C) の方針は

"任意に与えられた模様に対応する各行をブロックにして，入力として10進数を入れ，内部で p 進数に変換することにより，変換されたこの p 進数が，ちょうどブロックごとの模様を印刷するように数値と記号とを対応づける"

ことである。

各記号ごとに印刷する色の指定を行なえば，色彩をもつ模様の印刷も自由に行なえるのが，この方法のもつ一つの利点であろう。

3°. 各種の模様の作成

実際には，FACOM 230-45S を利用してこれら3種のアルゴリズム (A), (B), (C) の作成方針にのっとって，各種のプログラムを作成した。以下に方法ごとに分類してこれらを説明しよう。

3.1. アルゴリズム (A) による方法

アルゴリズム (A) は模様に応じてサブルーチンを作成し，その組合わせにより新しい模様

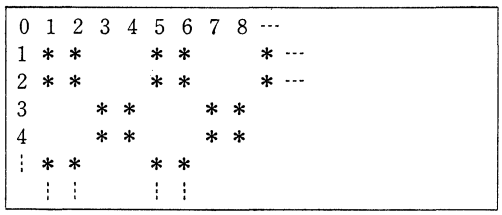
のサブルーチンを合成する方法であるから，サブルーチンの開発ということが一つの焦点である。

3.1.1. 市松模様

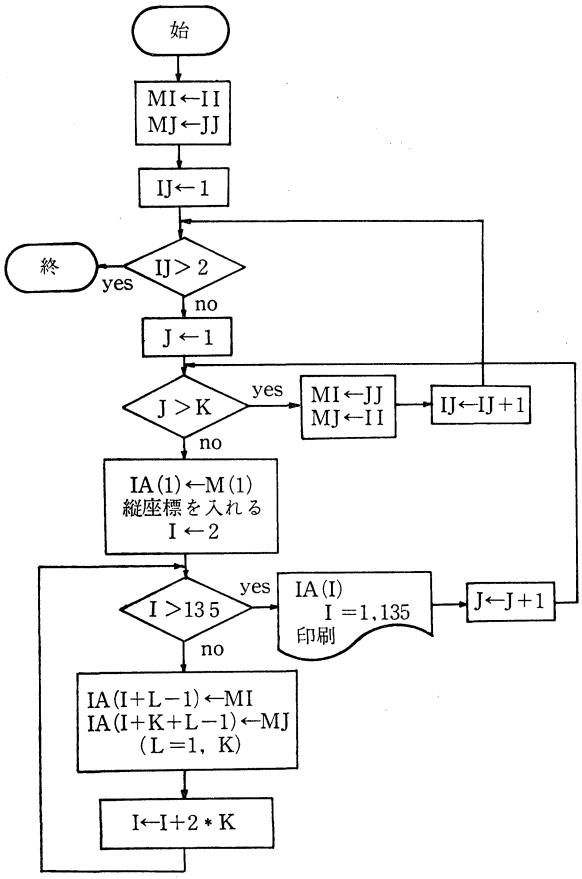
市松模様の中心となるのは次のサブルーチン

ITIMAT (M, II, JJ, K, KK, ISP)

である。ここに，Mは1次元の配列で，M(I)は一頁のたて座標を示す。II, JJは模様に使用する記号（たとえばII＝空，JJ＝＊のように任意に指定する），Kは市松模様の大きさである。たとえば，第1図のような市松模様を作成するにはK=2ととる。Kの値を自由に変更することにより，模様の大きさを希望に応じて変更できる。KKはたての大きさを制御する変数で，ISPはサブルーチンの使用を中断するための変数である。サブルーチンITIMATの構成を流れ図に書けば，第2図のようになる。



第1図　市松模様



第2図　ITIMAT の流れ図

3.1.2. 連点模様

市松模様の作成で気付いたことは，各行ごとの記号列を印刷するサブルーチンを用意して，パラメータの変更により各行，あるいは数行ごとに，模様に変化をつけることができることである。この各行ごとの模様形式を簡単に連点模様と名づけよう。連点模様は模様の基本構成である。この組合わせでいろいろな模様がつくれる。

サブルーチンは

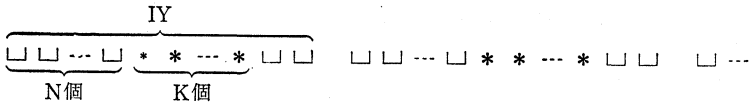
RENTEN(M, K1, K2, IY, N, K, IZ)

である。ここにK1, K2は模様に使用する記号変数（たとえばK1＝空，K2＝＊），IYは模様の横の大きさ，Nは記号K1のひきつづいて現われる最初の部分（第3図参照）の長さ（≧0），Kは記号K2のひきつづいて現われる個数（≧1）を指す。IZはたて座標を制御する変数である。

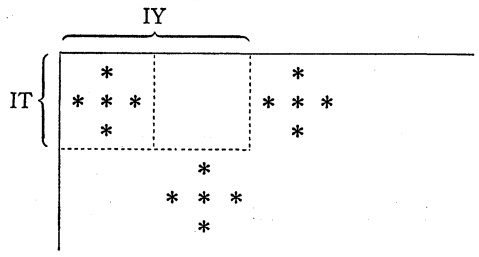
この連点模様を最初に使用して成功したのは次の十字模様である。

3.1.3. 十字模様

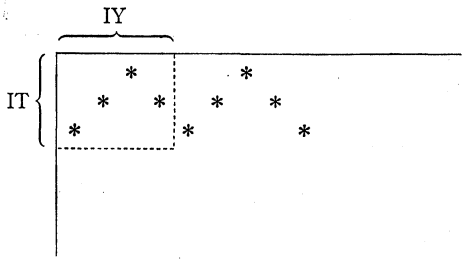
サブルーチンは



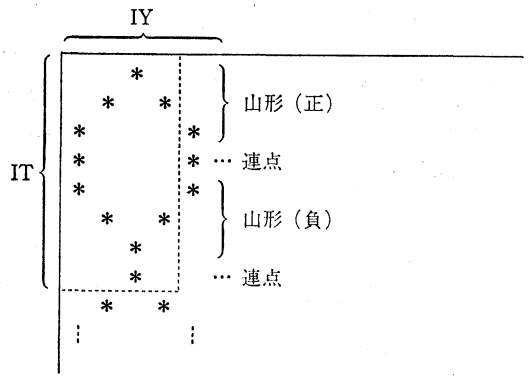
第3図　連点模様



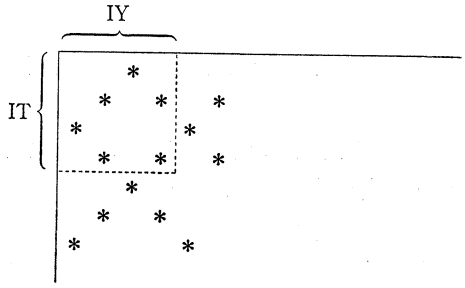
第4図　十字模様



第5図　山形模様（K=3 のとき）



第6図　亀甲模様



第7図　菱形模様（K=3）

JUJI (M, K1, K2, IY, IT, IZ, KG, K)

である。M, K1, K2, IY, IZ, K は前述の RENTEN の場合と同じで，IZ は模様のたての大きさ，KG はサブルーチンを中断する変数である。(第4図参照)

第4図は K=3 の場合である。K を変更することにより，模様の大きさを自由に制御できる。

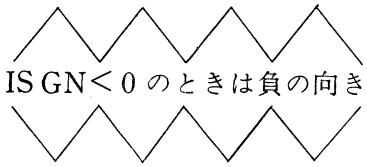
以下の各種模様は，上と同様にしてサブルーチンの構成が主要話題であるが，原理はまったく同じであるから，簡単に模様の図式とサブルーチンとを列挙することにしよう。

3.1.4.　山形模様

サブルーチンは

YAMA (M, K1, K2, IY, IT, IZ, KG, IS GN)

であり，M, K1, K2, IY, IT, IZ, KG は前と同様な意味をもつパラメータ，IS GN は山形の向きを示す変数で，IS GN≧0 のときは正の向き



を示す。

3.1.5.　亀甲模様

サブルーチン

KIKO (M, K1, K2, Y, IT, IZ, KG, K)

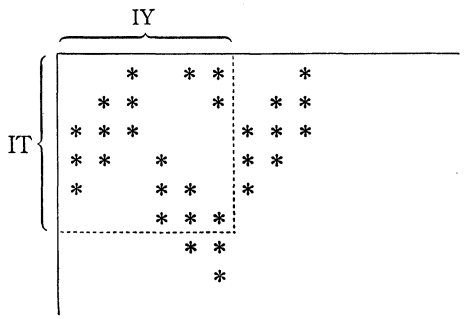
は第6図に示したように，2つのサブルーチン，RENTEN と YAMA とを組み合わせて構成する。

ここに K は亀甲模様の一辺の大きさである。

3.1.6.　菱形模様

サブルーチンは

HISHIG (M, K1, K2, IY, IT, IZ, KG, K)



第8図　矢絣模様

でKは菱形の一辺の大きさである。

3.1.7.　矢絣模様

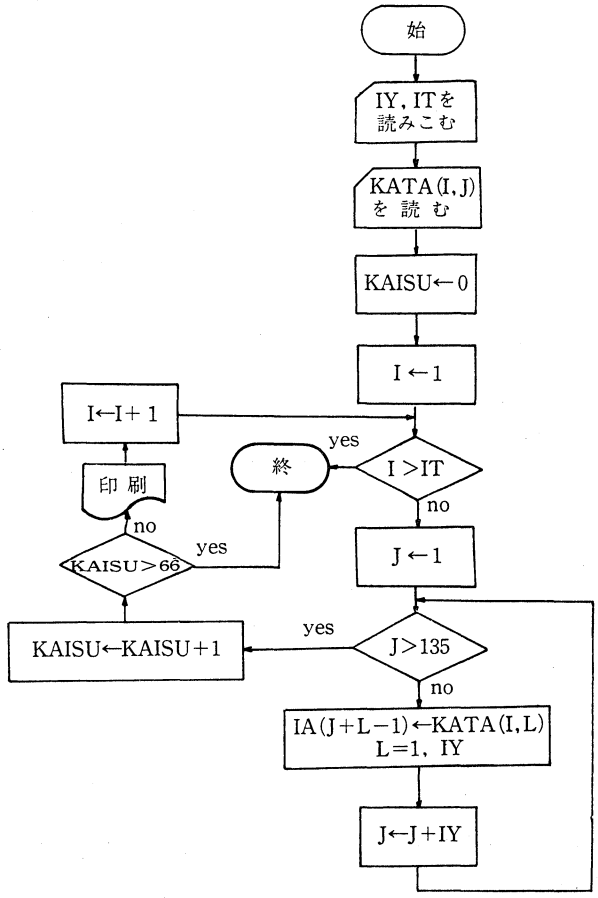
サブルーチンはRENTENをくり返し使用して構成された。

YAGASU(M, K1, K2, IY, IT, IZ, KG)

である。

3.2.　アルゴリズム（B）による方法

模様を文字形式でデータカードより入力して作成する方法である。KATA（I, J）を模様のデータを格納する2次元配列とするとき，第9図の流れ図でこの方法は構成することができる。実際例では矢絣模様を作成してみたが，プログラムの長さが（A）の方法に比し極度に短かくできる。



第9図　アルゴリズム（B）の方法の流れ図：

3.3.　アルゴリズム（C）による方法

模様を10進数に表現して，データカードより入力してN進数で模様化する方法である。その構成の主要な部分はサブルーチン

DESIGN (IY, IT, N, NUT, K)

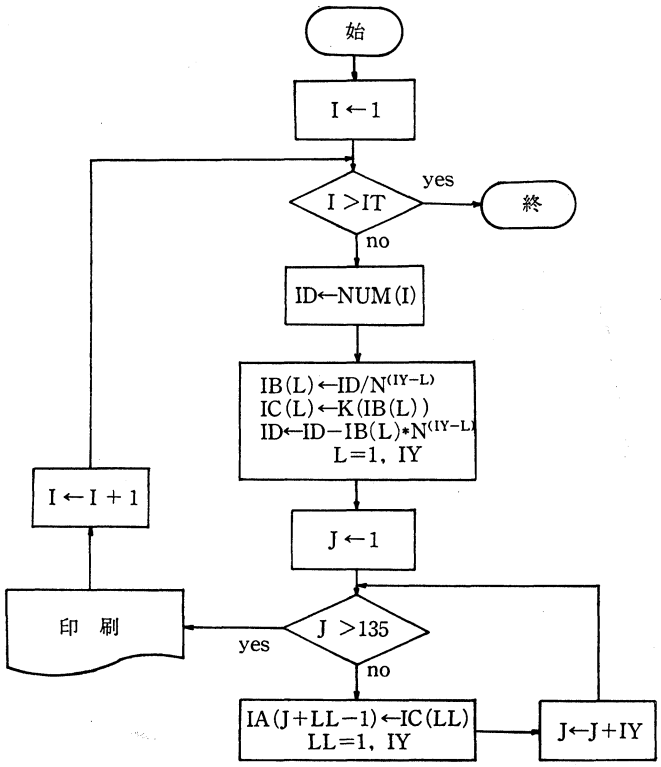
である。ここにNはN進法を示すN, NUM(I)は10進数のデータの格納されている1次元配列，K(I)は模様を構成するN個の文字の格納されている1次元配列である。流れ図は第10図のようになる。実際にプログラミングしたのは2例で，矢絣の印刷式をそれぞれ変更してみた。

4°.　結　　論

以上のプログラミングの結果，3種のアルゴリズムの優劣比較ができて，次のようなことが判明した。

アルゴリズム（A）による方法は次のような長所をもつ

(1A)　サブルーチンを使用して，指定された模様を希望通りの大きさに印刷できる。



第10図 アルゴリズム (C) の方法による流れ図

(2A) 模様を組合わせることにより，新しい模様を合成できる。
(3A) データカードを必要としない。
(4A) 模様に幾何学的な規則性があるものには特に適している。
一方，短所もないわけではない。とくに
(5A) 複雑な模様のプログラムは困難である。
(6A) 模様の指定の変化するたびに新しく作りかえなければならない。
などは欠点であろう。

アルゴリズム (B) による方法の長所は
(1B) プログラムが1つでよい。あとは，データの入れかえだけでどんな複雑な模様でも容易に印刷できる。
(2B) (A) の場合に比して，プログラムが非常に簡単になる。
(3B) 文字形式のデータを使用するので，パターンが鮮明である。
等である。この方法では
(4B) 同じ模様の場合でも大きさが変わるたびにデータを入れかえねばならない。
という欠点があるが，そんなに重大な欠点であるとは思えない。

最後にのべたアルゴリズム (C) による方法は (A), (B) にまさる方法で，
(1C) プログラムを1つ作成しておけば，データの変更だけで，どんな複雑な模様でも容易に印刷できる。
(2C) (A), (B) の方法に比較して，プログラムがもっとも簡単に作成される。
(3C) 10進数表示のデータを取扱うのでデータの取扱いが便利である。

短所は (B) の方法と共通するが，これはそんなに重大な欠点であるとはいえないであろう。

5°. プログラムと結果

以下に代表的なプログラムと結果をかかげよう。配列は3節の順序による。

```
      * FORTRAN  PRINT,00013
      C     ITIMATSU
      C
            DIMENSION  IA(140),M(10)
            READ(5,100) (M(I),I=1,10),II,JJ
        100 FORMAT(12A1)
            WRITE(6,102)
        102 FORMAT(1H1)
            READ(5,101) K
        101 FORMAT(I5)
            K2=2*K
            IS=(65/K2)*K2
            KAISU=0
            DO 10 I=1,135,10
            DO 10 J=1,10
         10 IA(I+J-1)=M(J)
            WRITE(6,201) (IA(I),I=1,135)
        201 FORMAT(1H+,135A1)
            MK=M(1)
            DO 20 I=1,9
         20 M(I)=M(I+1)
            M(10)=MK
         60 IF(KAISU.GE.IS) GO TO 70
            CALL ITIMAT(M,II,JJ,K,KAISU,K2)
            GO TO 60
         70 IT=65-IS
            CALL ITIMAT(M,II,JJ,K,KAISU,IT)
            STOP
            END
            SUBROUTINE ITIMAT(M,II,JJ,K,KK,ISP)
            DIMENSION IA(140),M(10)
            MARK=0
            MI=II
            MJ=JJ
            DO 30 IJ=1,2
            DO 20 J=1,K
            MARK=MARK+1
            IF(MARK.GT.ISP) GO TO 99
            IA(1)=M(1)
            K2=2*K
            DO 10 I=2,135,K2
            DO 12 L=1,K
         12 IA(I+L-1)=MI
            DO 13 L=1,K
         13 IA(I+K+L-1)=MJ
         10 CONTINUE
            WRITE(6,200) (IA(I),I=1,135)
        200 FORMAT(1H ,135A1)
            MK=M(1)
            DO 11 I=1,9
         11 M(I)=M(I+1)
            M(10)=MK
         20 CONTINUE
            MM=MI
            MI=MJ
            MJ=MM
         30 CONTINUE
            KK=KK+2*K
         99 RETURN
            END
      */
      0123456789*
          2
      * JEND
```

1. 市松模様プログラムとデータ (K=2)

```
01234567890123456789012345678901234567890
1**  **  **  **  **  **  **  **  **  **
2**  **  **  **  **  **  **  **  **  **
3  **  **  **  **  **  **  **  **  **  **
4  **  **  **  **  **  **  **  **  **  **
5**  **  **  **  **  **  **  **  **  **
6**  **  **  **  **  **  **  **  **  **
7  **  **  **  **  **  **  **  **  **  **
8  **  **  **  **  **  **  **  **  **  **
9**  **  **  **  **  **  **  **  **  **
0**  **  **  **  **  **  **  **  **  **
1  **  **  **  **  **  **  **  **  **  **
2  **  **  **  **  **  **  **  **  **  **
3**  **  **  **  **  **  **  **  **  **
4**  **  **  **  **  **  **  **  **  **
5  **  **  **  **  **  **  **  **  **  **
6  **  **  **  **  **  **  **  **  **  **
7**  **  **  **  **  **  **  **  **  **
8**  **  **  **  **  **  **  **  **  **
9  **  **  **  **  **  **  **  **  **  **
```

2. 市松模様(1の出力) ↑A(K=2) B(K=1,5)→

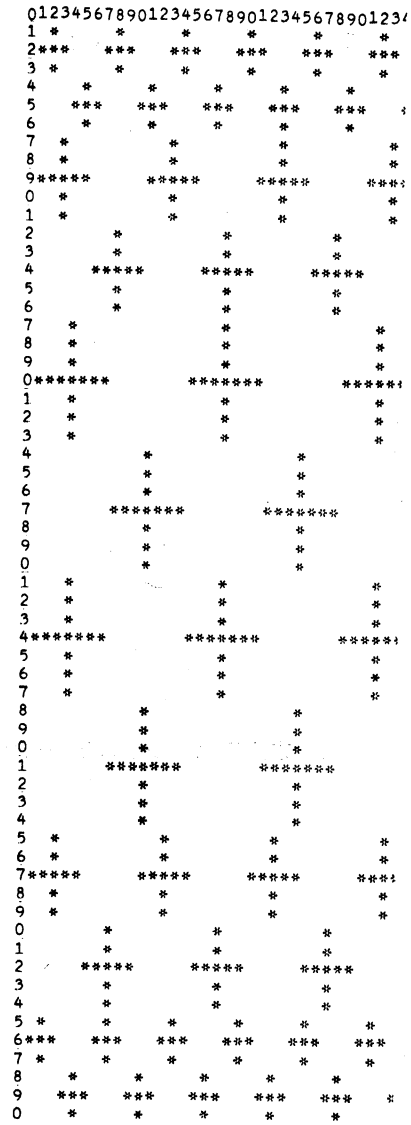
```
01234567890123456789012345678901234567890
1 *     *     *     *     *     *     *
2***   ***   ***   ***   ***   ***   ***
3 *     *     *     *     *     *     *
4   *     *     *     *     *     *
5  ***   ***   ***   ***   ***   ***   *
6   *     *     *     *     *     *
7 *     *     *     *     *     *     *
8***   ***   ***   ***   ***   ***   ***
9 *     *     *     *     *     *     *
0   *     *     *     *     *     *
1  ***   ***   ***   ***   ***   ***   *
2   *     *     *     *     *     *
3 *     *     *     *     *     *     *
4***   ***   ***   ***   ***   ***   ***
5 *     *     *     *     *     *     *
6   *     *     *     *     *     *
7  ***   ***   ***   ***   ***   ***   *
8   *     *     *     *     *     *
9 *     *     *     *     *     *     *
```

3.十 字 模 様 (K=3)

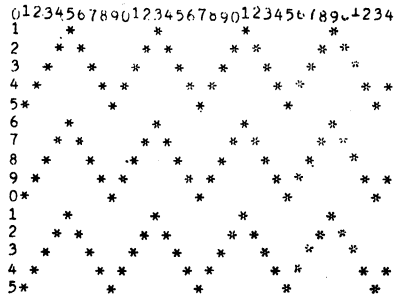
```
01234567890123456789012345678901234
1* * * * * * * * * * * * * * * * * *
2 * * * * * * * * * * * * * * * * * *
3**  **  **  **  **  **  **  **  **
4**  **  **  **  **  **  **  **  **
5  **  **  **  **  **  **  **  **
6  **  **  **  **  **  **  **  **
7***   ***   ***   ***   ***   ***
8***   ***   ***   ***   ***   ***
9***   ***   ***   ***   ***   ***
0   ***   ***   ***   ***   ***
1   ***   ***   ***   ***   ***
2   ***   ***   ***   ***   ***
3****    ****    ****    ****    **
4****    ****    ****    ****    **
5****    ****    ****    ****    **
6****    ****    ****    ****    **
7    ****    ****    ****    ****
8    ****    ****    ****    ****
9    ****    ****    ****    ****
0    ****    ****    ****    ****
1*****   *****   *****   *****   ***
2*****   *****   *****   *****   ***
3*****   *****   *****   *****   ***
4*****   *****   *****   *****   ***
5*****   *****   *****   *****   ***
6     *****   *****   *****
7     *****   *****   *****
8     *****   *****   *****
9     *****   *****   *****
0     *****   *****   *****
1******  ******  ******  ******
2******  ******  ******  ******
3******  ******  ******  ******
4******  ******  ******  ******
5******  ******  ******  ******
6******  ******  ******  ******
7      ******  ******  ******  ***
8      ******  ******  ******  ***
9      ******  ******  ******  ***
0      ******  ******  ******  ***
1      ******  ******  ******  ***
2      ******  ******  ******  ***
3****    ****    ****    ****    **
4****    ****    ****    ****    **
5****    ****    ****    ****    **
6****    ****    ****    ****    **
```

```
C      TOOSHI-MOYO (JUJI,K=3,9,2---7,3,-2)
       DIMENSION M(10)
       READ(5,100) (M(I),I=1,10),II,JJ
  100  FORMAT(12A1)
       WRITE(6,101)
  101  FORMAT(1H1)
       KAISU=0
       CALL ZAHYO(M)
       DO 10 I=3,7,2
       IY=2*I
   10  CALL JUJI(M,II,JJ,IY,IY,KAISU,IY,I)
       DO 20 I=3,7,2
       J=10-I
       IY=2*J
   20  CALL JUJI(M,II,JJ,IY,IY,KAISU,IY,J)
       STOP
       END
C          ****  (K=3)
C      M...ZAHYO NO MEMORI
C      K1..KIGO 1(*)
C      K2..KIGO 2( )
C      IY..YOKO MOYO NO OKISA
C      N...MAE  KIGO 2 NO OKISA
C      K...KIGO 1 NO OKISA
       SUBROUTINE RENTEN(M,K1,K2,IY,N,K,IZ)
       DIMENSION IA(200),M(10)
       IA(1)=M(1)
       DO 10 I=2,135
   10  IA(I)=K2
       DO 20 I=2,135,IY
       DO 20 L=1,K
   20  IA(I+N+L-1)=K1
       WRITE(6,200) (IA(I),I=1,135)
  200  FORMAT(1H ,135A1)
       MK=M(1)
       DO 30 I=1,9
   30  M(I)=M(I+1)
       M(10)=MK
       IZ=IZ+1
       RETURN
       END
C       *
C      ***     (K=3)
C       *
C      K...JUJI NO OKISA
       SUBROUTINE JUJI(M,K1,K2,IY,IT,IZ,KG,K)
       DIMENSION M(10)
       MARK=0
       KK=K/2+1
       DO 10 I=1,IT
       MARK=MARK+1
       IF(MARK.GT.KG) GO TO 99
       IF(I-K) 11,11,12
   11  IF(I.EQ.KK) GO TO 13
       IM=KK-I
       IN=1
       GO TO 15
   13  IM=0
       IN=K
       GO TO 15
   12  IF(I.EQ.K+KK) GO TO 14
       IM=K+KK-1
       IN=1
       GO TO 15
   14  IM=K
       IN=K
   15  CALL RENTEN(M,K1,K2,IY,IM,IN,IZ)
   10  CONTINUE
   99  RETURN
       END
       SUBROUTINE ZAHYO(M)
       DIMENSION M(10),IA(140)
       DO 10 I=1,135,10
       DO 10 J=1,10
   10  IA(I+J-1)=M(J)
       WRITE(6,201) (IA(I),I=1,135)
  201  FORMAT(1H+,135A1)
       MK=M(1)
       DO 20 I=1,9
   20  M(I)=M(I+1)
       M(10)=MK
       RETURN
       END
*/
0123456789*
* JEND
```

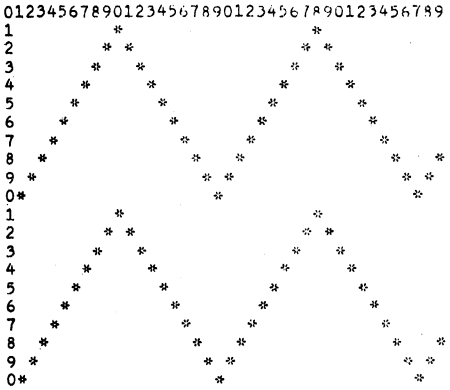
4. 十字模様プログラムとデータ



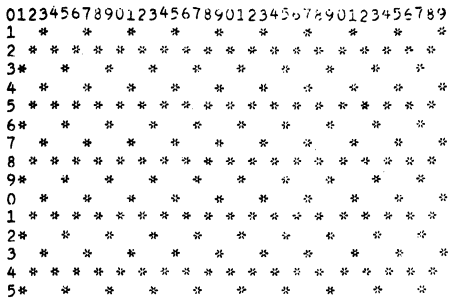
5. 十 字 模 様 (5の出力)



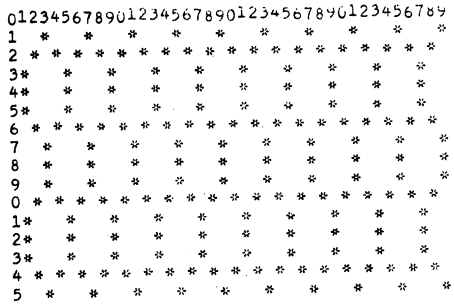
6. 山 形 模 様



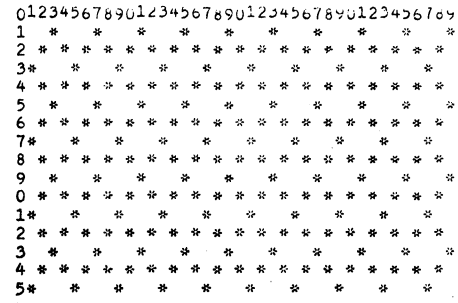
7. 山 形 模 様 (K=10)



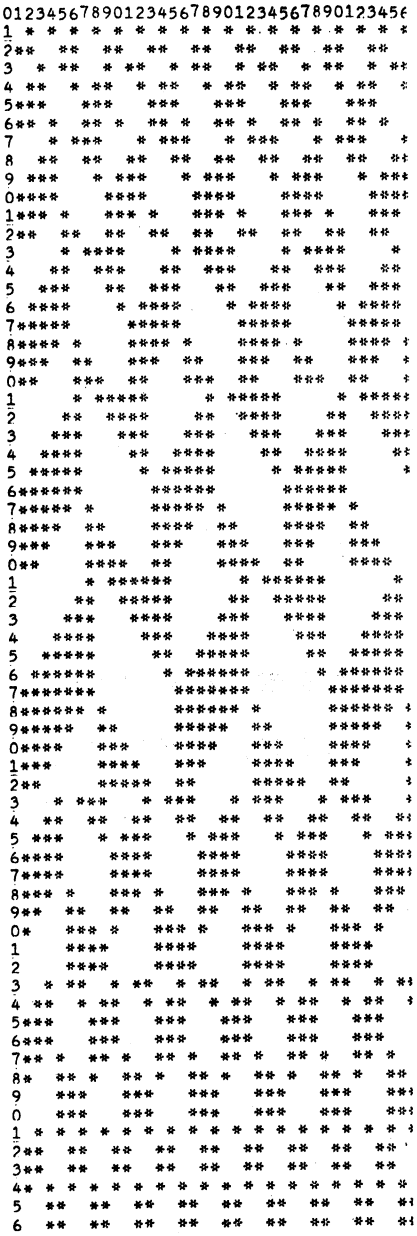
8. 山 形 模 様 (K=3)



9.亀 甲 模 様



10. 菱 形 模 様



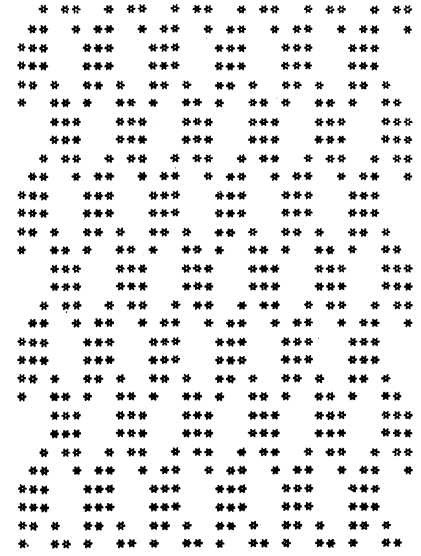
11. 矢 絣 模 様

```
C     MOYO O DATA CARD YERI INPUT
      DIMENSION KATA(10,10),IA(150)
      READ(5,100) K1,K2,IY,IT
  100 FORMAT(2A1,2I5)
      WRITE(6,999)
  999 FORMAT(1H1)
      DO 10 I=1,IT
   10 READ(5,101) (KATA(I,J),J=1,IY)
  101 FORMAT(10A1)
      KAISU=0
   25 DO 20 I=1,IT
      KAISU=KAISU+1
      IF(KAISU.GT.66) GO TO 99
      DO 30 J=1,135,IY
      DO 30 L=1,IY
   30 IA(J+L-1)=KATA(I,L)
      IF(KAISU.EQ.1) GO TO 15
      WRITE(6,200) (IA(J),J=1,135)
  200 FORMAT(1H ,135A1)
      GO TO 20
   15 WRITE(6,201) (IA(J),J=1,135)
  201 FORMAT(1H+,135A1)
   20 CONTINUE
      GO TO 25
   99 STOP
      END
*/
*     6    8
  * **
 **  *
***
***
** *
*  **
   ***
   ***
¥ JEND
```

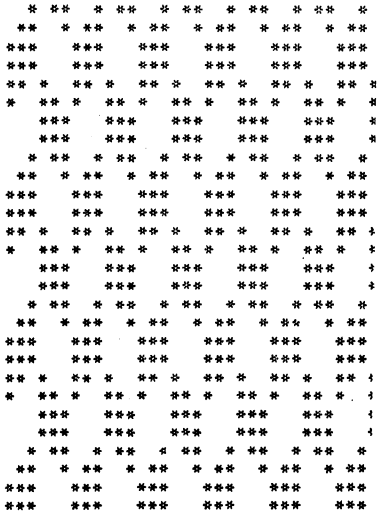
12. 矢絣模様プログラム (1) ↑

13. 矢絣模様プログラム (2) →

```
      DIMENSION NUM(20),K(10)
      READ(5,100) IY,IT,N
  100 FORMAT(3I5)
      WRITE(6,999)
  999 FORMAT(1H1)
      READ(5,101) (K(I),I=1,N)
  101 FORMAT(10A1)
      DO 10 I=1,IT
   10 READ(5,102) NUM(I)
  102 FORMAT(I15)
      KAISU=0
      IS=(65/IT)*IT
   60 IF(KAISU.GE.IS) GO TO 70
      CALL DESIGN(IY,IT,N,NUM,K)
      KAISU=KAISU+IT
      GO TO 60
   70 IO=65-IS
      CALL DESIGN(IY,IO,N,NUM,K)
      STOP
      END
      SUBROUTINE DESIGN(IY,IT,N,NUM,K)
      DIMENSION NUM(IT),K(10),IC(20),IB(20),IA(160)
      DO 20 I=1,IT
      ID=NUM(I)
      DO 30 L=1,IY
      IB(L)=ID/N**(IY-L)
      MK=IB(L)
      IC(L)=K(MK+1)
   30 ID=ID-IB(L)*N**(IY-L)
      DO 40 J=1,135,IY
      DO 40 LL=1,IY
   40 IA(J+LL-1)=IC(LL)
      WRITE(6,200) (IA(J),J=1,135)
  200 FORMAT(1H ,135A1)
   20 CONTINUE
      RETURN
      END
*/
    6    8    2
 *
           11
           25
           56
           56
           52
           38
            7
            7
¥ JEND
```



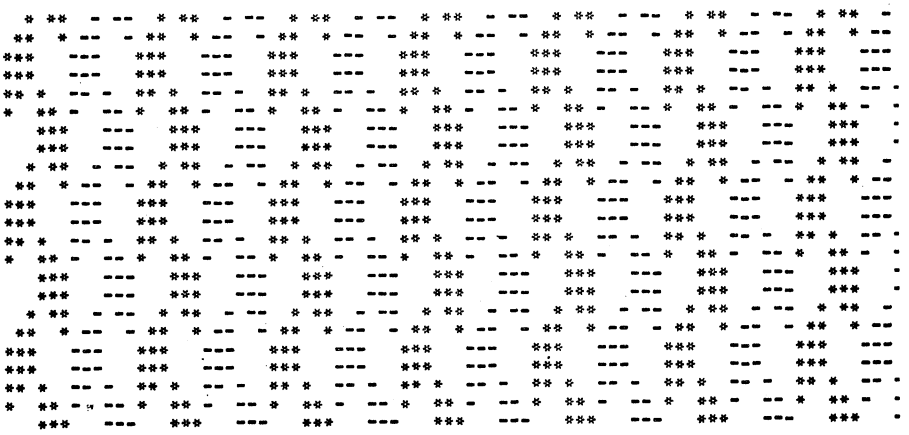
14. 矢絣模様 (12の出力)



15. 矢絣模様 (13の出力)

```
1         ¥ FORTRAN  PRINT,00013
2         C      MOYO NO DATA O SUDE INPUT
3         C      N...N-SHIN HYOJI
4               DIMENSION IA(160),NUM(20),IB(20),IC(20),K(10),KE(10)
5               READ(5,100) IY,IT,N
6           100 FORMAT(3I5)
7               WRITE(6,999)
8           999 FORMAT(1H1)
9               READ(5,101) (K(I),I=1,N)
10              READ(5,101) (KE(I),I=1,N)
11          101 FORMAT(10A1)
12              DO 10 I=1,IT
13           10 READ(5,102)  NUM(I)
14          102 FORMAT(I15)
15              KAISU=0
16           25 DO 20 I=1,IT
17              ID=NUM(I)
18              DO 30 L=1,IY
19              IB(L)=ID/N**(IY-L)
20              MK=IB(L)
21              IC(L)=K(MK+1)
22              IC(IY+L)=KE(MK+1)
23           30 ID=ID-IB(L)*N**(IY-L)
24              KAISU=KAISU+1
25              IF(KAISU.GT.66) GO TO 99
26              IY2=2*IY
27              DO 40 J=1,135,IY2
28              DO 40 LL=1,IY2
29           40 IA(J+LL-1)=IC(LL)
30              IF(KAISU.EQ.1) GO TO 15
31              WRITE(6,200) (IA(J),J=1,135)
32          200 FORMAT(1H ,135A1)
33              GO TO 20
34           15 WRITE(6,201) (IA(J),J=1,135)
35          201 FORMAT(1H+,135A1)
36           20 CONTINUE
37              GO TO 25
38           99 STOP
39              END
40        */
41             6    8    2
42         *
43         -
44                        11
45                        25
46                        56
47                        56
48                        52
49                        38
50                         7
51                         7
52        ¥ JEND
```

16. 矢絣模様プログラム (3) とデータ



17. 矢 絣 模 様 (16の出力)