

## 14 配列 (array)

大量のデータを扱うとき、それぞれに変数名をつけると大変だし、個数が変わるとプログラムを作り直さないといけません。

同じ型の複数個のデータをまとめて1つの変数で表したものを配列 (array, 表のようなもの) といいます。

### 14.1 SosuuNoHyou.dpr (素数の表)

下記のプログラムは、エラトステネスの篩<sup>ふるい</sup>法によって  $n_{\max}$  以下の素数をすべて求めるものです。最終的に、次の表が作られますが、その途中経過も表示するようにしています。

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0+ :	1	1	0	0	2	0	2	0	2	3
10+ :	2	0	2	0	2	3	2	0	2	0
20+ :	2	3	2	0	2	5	2	3	2	0
30+ :	2	0	2	3	2	5	2	0	2	3
40+ :	2	0	2	0	2	3	2	0	2	7
50+ :	2	3	2	0	2	5	2	3	2	0
60+ :	2	0	2	3	2	5	2	0	2	3
70+ :	2	0	2	0	2	3	2	7	2	0
80+ :	2	3	2	0	2	5	2	3	2	0
90+ :	2	7	2	3	2	5	2	0	2	3

この表の意味は、たとえば 60+ 行目の +7 列目の 0 が “67 は素数” を示しています。また、90+ 行目の +1 列目の 7 が “91 は合成数で最小素因数は 7” を示しています。このように、 $n$  番目の欄に 0 が書いてあったら “ $n$  は素数”，2 以上の数  $p$  が書いてあったら “ $n$  は合成数で最小素因数は  $p$ ” を意味しています。 $n = 0, 1$  は素数でも合成数でもありません。

```

1 program SosuuNoHyou; // 学生証番号 氏名
2 {$APPTYPE CONSOLE}
3 uses SysUtils;
4
5 const
6   NMax = 99;
7 type
8   THyou = array [0..NMax] of Integer;
9
10 var
11   SosuuTable : THyou;
12
13 procedure WriteTable;
14   { 素数の表を書いて、Enter キーを押すまで待つ }
15   var
16     N : Integer;
17   begin
18     Write('':6);
19     N := 0; // 見出し行を書く
20     repeat
21       Write('+'':6, N);
22       Inc(N);
23     until N > 9;
24     WriteLn;

```

```

25     N := 0;
26     repeat
27         if N mod 10 = 0                                // 一の位が0のとき
28             then begin
29                 WriteLn;
30                 Write(N:3, '+ :');                    // 見出しを書く
31             end;
32         Write(SosuuTable[N]:7);                        // 表の中身を書く
33         Inc(N);
34     until N > NMax;
35     WriteLn;
36     WriteLn('push Enter key':78);
37     ReadLn;
38 end; {WriteTable}
39
40 procedure Erathostenes;
41     { エラトステネスの篩 (ふるい) 法により素数と合成数を区別する }
42     { nが素数      SosuuTable[n] = 0                    }
43     { nが合成数    SosuuTable[n] = nの最小素因数        }
44     { nが0, 1     SosuuTable[n] = 1                    }
45 var
46     N, Ns : Integer;
47 begin
48     SosuuTable[0] := 1;
49     SosuuTable[1] := 1;
50     N := 2;                                           // すべて0 (素数候補) に初期設定
51     repeat
52         SosuuTable[N] := 0;
53         Inc(N);
54     until N > NMax;
55     WriteTable;
56     N := 2;
57     repeat
58         WriteLn(N, 'について実行');
59         if SosuuTable[N] = 0                            // まだ0のまま, すなわちnが素数のとき
60             then begin
61                 Ns := Sqr(N);                            // n^2以上のnの倍数を素数候補から除く
62                 repeat
63                     if SosuuTable[Ns] = 0                // まだ0のまま, すなわちnが最小素因数
64                         then SosuuTable[Ns] := N;
65                     Ns := Ns+N;
66                 until Ns > NMax;
67                 WriteTable;
68             end;
69         Inc(N);
70     until Sqr(N) > NMax;
71 end; {Erathostenes}
72
73 begin {Main}
74     Erathostenes;
75     WriteLn('終了');
76     ReadLn;
77 end.

```

注 14.1 画面では10個毎に改行して10行10列の2次元の表になっていますが、計算機内部では100個の数を一列に並べた1次元の表です。

## 14.2 問題

- (1) これを実行して途中経過を見てエラトステネスの篩法を理解してください。
- (2) 6 行目で定数 NMax を定義しています。これを 999 に変えて実行しなさい。“終了”となるまでに、Enter キーを何回押すでしょうか。押した回数が、999 までの素数の個数です。
- (3) 途中経過を書かないように Erathostenes の中の出力関係の行 (3 行) を削除し、最後に完成した表を書くように 75 行目を変更しなさい。
- (4) 下記のような表 (ただし、999 まで) を書くように WriteTable を変更しなさい。

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0+ :	1*0	1*1	素数	素数	2*2	素数	2*3	素数	2*4	3*3
10+ :	2*5	素数	2*6	素数	2*7	3*5	2*8	素数	2*9	素数
20+ :	2*10	3*7	2*11	素数	2*12	5*5	2*13	3*9	2*14	素数
30+ :	2*15	素数	2*16	3*11	2*17	5*7	2*18	素数	2*19	3*13
40+ :	2*20	素数	2*21	素数	2*22	3*15	2*23	素数	2*24	7*7
50+ :	2*25	3*17	2*26	素数	2*27	5*11	2*28	3*19	2*29	素数
60+ :	2*30	素数	2*31	3*21	2*32	5*13	2*33	素数	2*34	3*23
70+ :	2*35	素数	2*36	素数	2*37	3*25	2*38	7*11	2*39	素数
80+ :	2*40	3*27	2*41	素数	2*42	5*17	2*43	3*29	2*44	素数
90+ :	2*45	7*13	2*46	3*31	2*47	5*19	2*48	素数	2*49	3*33

[ヒント]  $n = p \times q$  のとき、 $q$  が 1 桁 (0~9)、2 桁 (10~99)、3 桁 (100~999) の場合に分けて (case 文を使って)  $p$  を何文字で書くか (Write(P: )の ) 決めます。

- (5) 素数の表を書いた後で、素数だけを抜き出して書くようにしなさい。下記のように (ただし、997 まで) 1 行に 10 個ずつきれいに出力すること。

2	3	5	7	11	13	17	19	23	29
31	37	41	43	47	53	59	61	67	71
73	79	83	89	97					

## 14.3 問題

```
N := 開始値;
repeat
  文たち
  Inc(N);
until N > 終了値
```

の形の repeat 文は、「 $n =$  開始値, 開始値 + 1, ..., 終了値 について繰り返し実行する」ためのものですが、これは次のように for 文に変えることができます。

```
for N := 開始値 to 終了値 do
  begin
    文たち
  end
```

SosuuNoHyou.dpr の repeat 文はすべてこの形をしているので、for 文に書き換えなさい。

[注]  $\text{Sqr}(N) < \text{NMax}$  は  $N < \text{Trunc}(\text{Sqrt}(\text{NMax}))$  と同じ。