

課題 素数の生成

正整数 n 以下の素数をすべて求める問題を考察する。
 まず、素朴な方法を取り上げる。つぎに、「ふるい」という考え方を適用すると、劇的に改良されることを示す。

方法 1 素朴な方法

素数の候補（3以上の奇数） p を用意し、その候補に対して、3以上 $p-2$ 以下の奇数 d で割っていく。どの数でも割り切れない場合、素数とし、いずれかの数で割り切れた場合、合成数とする。

素数の候補 p	3, 5, 7, 9, 11, 13, 15, 17, 19, …
除数 d	奇数で、 $3 \leq d \leq p-2$

●例

25 は、3で割り切れないが、5で割り切れるので合成数である。
 29 は、3, 5, 7, …, 27で割り切れないので素数である。

●プログラム (PR111. bas)

```

1  ' << PR111. bas >>
2  ' 素数の生成 (方法 1 : 素朴な方法)
3  '
4  Dim A(10000): ' 素数を保存する配列。
5  '
6  Do
7  ' 正整数Nの読み込み。
8  Read N
9  If (N <= 0) or (N > 10000) Then Exit Do
10 '
11 ' 初期設定。2を素数として保存。
12 K=1: A(K)=2
13 ' P:素数候補の初期値。
14 P=3:
15 '
16 ' 素数の探索。
17 While P <= N
18   D=3: ' 除数の初期値。
19   While D <= P-2
20     If P Mod D <> 0 Then
21       ' 割り切れない場合。
22       D=D+2: ' 除数の更新。
23     Else
24       ' 割り切れた場合。
25       Exit Do
26     End If
27   Wend

```

```

28     '
29     If D >= P Then
30         ' 素数候補Pを素数として保存。
31         K=K+1: A(K)=P
32     End If
33     ' 素数候補Pの更新。
34     P=P+2
35 Wend
36
37 ' 求めた素数の表示。
38 For I=1 To K
39     Print Using"#####";A(I);
40     If I Mod 10 = 0 Then Print
41 Next I
42 Print
43 Print N;"以下の素数 : ";K;"個"
44 Loop
45 End
46 '
47 ' データ。
48 Data 1000, 0

```

実行結果

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	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997		

1000以下の素数 : 168個
OK

方法2 方法1の改良

改良点：方法1において、除数dの範囲を小さくする。

素数の候補（3以上の奇数）pを用意し、その候補に対して、3以上、 $[\sqrt{p}]$ 以下の奇数dで割っていく。どの数でも割りきれない場合、素数とし、いずれかの数で割り切れた場合、合成数とする。ここで、 $[x]$ は実数xの整数部分を意味する。

素数の候補 p	3, 5, 7, 9, 11, 13, 15, 17, 19, ...
除数 d	奇数で、 $3 \leq d \leq [\sqrt{p}]$

●例

25 は、5で割り切れるので合成数である。
29 は、3, 5で割り切れないので素数である。

● $[\sqrt{p}]$ 以下の奇数dで割ればよい理由

$[\sqrt{p}]$ より大きい奇数dで割り切れたとすると、 p/d は $[\sqrt{p}]$ 以下の奇数となる。すなわち、 $[\sqrt{p}]$ 以下の奇数の約数があることになる。つまり、3以上 $[\sqrt{p}]$ 以下の奇数で割って調べれば十分である。

●プログラム (PR211. bas)

```

1  ' << PR211. bas >>
2  ' 素数の生成（方法1の改良）
3  '
4  Dim A(10000): ' 素数を保存する配列。
5  '
6  Do
7  ' 正整数Nの読み込み。
8  Read N
9  If (N <= 0) or (N > 10000) Then Exit Do
10 '
11 ' 初期設定。2を素数として保存。
12 K=1: A(K)=2
13 ' P:素数候補の初期値。
14 P=3:
15 '
16 ' 素数の探索。
17 While P <= N
18   D=3: ' 除数の初期値。
19   While          <= P
20     If P Mod D <> 0 Then
21       ' 割り切れない場合。
22       D=D+2: ' 除数の更新。
23     Else
24       ' 割り切れた場合。
25     Exit Do

```

```
26     End If
27   Wend
28   '
29   If  Then
30     ' 素数候補Pを素数として保存。
31     K=K+1: A(K)=P
32   End If
33   ' 素数候補Pの更新。
34   P=P+2
35 Wend
36
37 ' 求めた素数の表示。
38 For I=1 To K
39   Print Using"#####";A(I);
40   If I Mod 10 = 0 Then Print
41 Next I
42 Print
43 Print N;"以下の素数 : ";K;"個"
44 Loop
45 End
46 '
47 ' データ。
48 Data 1000, 0
```

実行結果：方法1と同じ

方法3 エラトステネスの方法

正整数 n 以下の素数をつぎの手順で生成する。

- 手順 (0) 2 を素数とする。
 手順 (1) 数列 $\{3, 5, 7, \dots, n\}$ を用意する。
 手順 (2) 数列の中の最小値 p を素数とし、その数自身とその数の**倍数** ($2p, 3p, 4p, \dots$) を数列から取り除き、新しい数列を構成する。
 手順 (3) 新しい数列が空になるまで手順 (2) を繰り返す。

●例 $n=29$

2を素数とする。

- 1 番目の数列 $\{3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29\}$
 3を素数とし、3の倍数を削除する。

- 2 番目の数列 $\{5, 7, 11, 13, 17, 19, 23, 25, 29\}$
 5を素数とし、5の倍数を削除する。

- 3 番目の数列 $\{7, 11, 13, 17, 19, 23, 29\}$
 7を素数とし、7の倍数を削除する。

- 4 番目の数列 $\{11, 13, 17, 19, 23, 29\}$
 11を素数とし、11の倍数を削除する。

- 5 番目の数列 $\{13, 17, 19, 23, 29\}$
 13を素数とし、13の倍数を削除する。

- 6 番目の数列 $\{17, 19, 23, 29\}$
 17を素数とし、17の倍数を削除する。

- 7 番目の数列 $\{19, 23, 29\}$
 19を素数とし、19の倍数を削除する。

- 8 番目の数列 $\{23, 29\}$
 23を素数とし、23の倍数を削除する。

- 9 番目の数列 $\{29\}$
 29を素数とし、29の倍数を削除する。

- 10 番目の数列 $\{\}$
 空集合となり終了。

●プログラム (PR311. bas)

```

1  ' << PR311. bas >>
2  ' 方法3 : エラトステネスの方法
3  '
4  Dim X(10000): ' 数列を配列Xで表現する。
5                ' すなわち、数列の要素Iは、X(I)=0、
6                ' 数列の要素でなくなると、X(I)=1とする。
7  '
8  Do
9    ' 正整数Nの読み込み。
10   Read N
11   If (N <= 0) or (N > 10000) Then Exit Do
12   '
13   ' 初期設定。Iが素数の候補である間、X(I)=0。
14   '           Iが素数pの倍数となる場合、X(I)=1。
15   For I=3 To N: X(I)=0: Next I
16   ' 素数候補Pの初期値。
17   P=3
18   '
19   ' 素数の探索。
20   While 
21     If X(P) = 0 Then
22       ' Pの倍数を削除。
23       For K=2*P To N Step P: X(K)=1: Next K
24     End If
25     '
26     ' 素数候補Pの更新。
27     P=P+2
28   Wend
29   '
30   ' X(I)=0 となる数Iは素数なので表示。
31   COUNT=1: ' 素数の個数。
32   Print Using"#####";2;
33   For I=3 To N Step 2
34     If X(I) = 0 Then
35       COUNT=COUNT+1: Print Using"#####";I;
36       If COUNT Mod 10 = 0 Then Print
37     End If
38   Next I
39   Print
40   Print N;"以下の素数 : ";COUNT;"個"
41 Loop
42 End
43 '
44 ' データ。
45 Data 1000,0

```

実行結果

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	101	103	107	109	113
127	131	137	139	149	151	157	163	167	173
179	181	191	193	197	199	211	223	227	229
233	239	241	251	257	263	269	271	277	281
283	293	307	311	313	317	331	337	347	349
353	359	367	373	379	383	389	397	401	409
419	421	431	433	439	443	449	457	461	463
467	479	487	491	499	503	509	521	523	541
547	557	563	569	571	577	587	593	599	601
607	613	617	619	631	641	643	647	653	659
661	673	677	683	691	701	709	719	727	733
739	743	751	757	761	769	773	787	797	809
811	821	823	827	829	839	853	857	859	863
877	881	883	887	907	911	919	929	937	941
947	953	967	971	977	983	991	997		

1000以下の素数： 168個
OK

方法4 エラトステネスの方法の改良

方法3を改良する。

$2*p, 3*p, 4*p, \dots, (p-1)*p$ は、すでに数列から取り除かれているはずだから省略できる。

- 手順 (0) 2を素数とする。
 手順 (1) 数列 $\{3, 5, 7, \dots, n\}$ を用意する。
 手順 (2) 数列の中の最小値 p を素数とし、その数自身とその数の**倍数** ($p*p, p*(p+1), p*(p+2), \dots$) を数列から取り除き、新しい数列を構成する。
 手順 (3) $p*p$ が n 以下の間、手順 (2) を繰り返す。

●例 $n=29$

2を素数とする。

- 1 番目の数列 $\{3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29\}$
 3を素数とし、 $3*3, 3*4, 3*5, 3*6, 3*7, 3*8, 3*9$ を削除する。

- 2 番目の数列 $\{5, 7, 11, 13, 17, 19, 23, 25, 29\}$
 5を素数とし、 $5*5$ を削除する。

- 3 番目の数列 $\{7, 11, 13, 17, 19, 23, 29\}$
 7を素数とする。削除する要素がなくなったので、 $\{11, 13, 17, 19, 23, 29\}$ を素数とし終了する。

●プログラム (PR411. bas)

```

1  ' << PR411. bas >>
2  ' 方法 4 : エラトステネスの方法の改良
3  '
4  Dim X(10000): ' 数列を配列Xで表現する。
5                ' すなわち、数列の要素Iは、X(I)=0、
6                ' 数列の要素でなくなると、X(I)=1とする。
7  '
8  Do
9    ' 正整数Nの読み込み。
10   Read N
11   If (N <= 0) or (N > 10000) Then Exit Do
12   '
13   ' 初期設定。Iが素数の候補である間、X(I)=0。
14   '           Iが素数pの倍数となる場合、X(I)=1。
15   For I=3 To N: X(I)=0: Next I
16   ' 素数候補Pの初期値。
17   P=3
18   '
19   ' 素数の探索。
20   While [ ]
21     If X(P) = 0 Then
22       ' Pの倍数を削除。
23       For K=2*P To N Step P: X(K)=1: Next K
24     End If
25     '
26     ' 素数候補Pの更新。
27     P=P+2
28   Wend
29   '
30   ' X(I)=0 となる数Iは素数なので表示。
31   COUNT=1: ' 素数の個数。
32   Print Using"#####";2;
33   For I=3 To N Step 2
34     If X(I) = 0 Then
35       COUNT=COUNT+1: Print Using"#####";I;
36       If COUNT Mod 10 = 0 Then Print
37     End If
38   Next I
39   Print
40   Print N;"以下の素数 : ";COUNT;"個"
41 Loop
42 End
43 '
44 ' データ。
45 Data 1000,0

```

実行結果 : 方法 3 と同じ