

繰り返し (while 文) (続き)

[例題 1] 入力された (2 以上の) 整数 n が素数であるかどうかを判定するプログラムを作りなさい。

(問題に対する一つの考え方) n が $2, 3, \dots, n-1$ のいずれでも割り切れなかったら n は素数である。従って、 n を $2, 3, \dots, n-1$ で順番に割っていき、割り切れた時にチェックしておく。そしてすべての割算が終了したあとでなにもチェックがされていなければ素数ということになる。

```
1  /* 素数の判定 */
2  #include <stdio.h>
3  main()
4  {
5      int i, n, check;
6
7      printf("number = ");
8      scanf("%d", &n);          /* 整数の入力 */
9
10     check = 0;                /* 割り切れたかどうかのチェック変数 初期値は 0 */
11     i = 2;                    /* n を 2, 3, 4, ... で割っていくが、最初の 2 */
12
13
14     while ( ) {              /* i が n 未満のあいだ繰り返す */
15
16
17         if ( ) {            /* n が i で割り切れるか? */
18
19             check = 1;      /* 割り切れたらチェック変数を 1 にセットする */
20         }
21         i = i + 1;          /* 次の i */
22     }
23
24     if (check == 0) {       /* もしチェック変数が 0 のままなら n は素数 */
25         printf("%d ha sosuu\n", n);
26     }
27     else {
28         printf("%d ha sosuu denai\n", n);
29     }
30 }
```

括弧内を埋めてプログラムを完成しなさい。わかったら、打ち込んで実行し、結果を確認しなさい。

[例題 2] 1000 までの素数をすべて求めなさい。

(問題の考え方) 2 から 1000 までの整数について一つずつ例題 1 の素数判定を行い、素数なら出力する。

```

1  /* 1000 までの素数 */
2  #include <stdio.h>
3  main()
4  {
5      int i, n, check;
6
7      printf("sosuu = ");
8      n = 2;                      /* 2, 3, ..., 1000 を順番に素数かどうか調べる */
9
10
11     while (          ) {        /* n が 1000 になるまで繰り返す */
12
13         check = 0;              /* 素数判定部は例題 1 と同じ */
14         i = 2;
15
16         while ( i < n ) {
17             if ( n % i == 0 ) {
18                 check = 1;
19             }
20             i = i+1;
21         }
22
23         if (check == 0) {
24             printf("%8d", n);    /* %8d は 8 文字分のスペースに整数を出力 */
25         }
26
27
28         ;                        /* 次の n */
29
30     }
31 }
```

(効率のよいプログラムに向けて)

上のプログラムは次のような点で効率のより良いものへ改良できる。

- 整数 n が \sqrt{n} 以下に約数 p を持つなら、 $\frac{n}{p}$ も約数である。すなわち、13 行目からの while 文で、 i は 2 から $n-1$ まででなく、 \sqrt{n} までで十分である。したがって、13 行目を `while (i * i <= n)` のように直してよい。
- もし n が $2 \leq m < n$ である m で割り切れるなら、さらに n が m より大きい数で割り切れるかどうかを調べることなく素数でないと言える。これをプログラムで実現するには、13 行目の while 文

の条件式中で i の大きさだけでなく、チェック変数の値も調べ、

```
while (i * i <= n && check == 0)
```

のように直せばよい。(または、後日学習する `break` 文を用いて強制的に `while` 文から抜ければよい。)

- また、 i が 2 から \sqrt{n} までといっても、このうち偶数は 2 のみで十分である。さらに、たとえば 3 で割り切れないことが判ったら、6 や 9 で割り切れるはずはない。すなわち、 n が \sqrt{n} 以下の素数で割り切れるかどうかのみを調べればよいことになる。このためには、求めた $n-1$ までの素数をどこかにためておいて、それを使って n が素数かどうかを調べればよい。これには後日学習する配列というものを使用することになる。
- n を 2 から 1000 まで 1 ずつ増やしているが、このうち素数となるのは 2 のみであるから 3 から先は 2 ずつ増やすことで、調べる数を半減することができる。

[例題 3] 次のプログラムは非負整数 m, n を読み込んで m^n を計算し、その値を出力するプログラムの一部である。括弧内を埋めてプログラムを完成しなさい。わかったら、打ち込んで実行し、結果を確認しなさい。

```

1  /* m の n 乗 */
2  #include <stdio.h>
3
4  main ()
5  {
6      int m, n, result;          /* m, n 及び結果をしまうための変数 result の宣言 */
7
8      scanf("%d %d", &m, &n);   /* m, n の読み込み */
9      result = 1;               /* n = 0 の時、m の 0 乗は 1 であるから、
10                                 始めは 1 を入れておく */
11
12
13     while (          ) {      /* n が 0 より大きい間、 */
14
15
16         result =          ;   /* result を m 倍したものを新しい result にし、 */
17
18         n = n - 1;           /* n を 1 減らすことを */
19     }                        /* 繰り返す */
20
21     printf("%d\n", result);   /* 結果を出力する */
22 }
```

[例題 4] 次のプログラムは例題 3 でやったことをより効率良く行うプログラムである¹。

プログラムを打ち込んで実行し、例題 3 の時と同じ結果をもたらすことを確かめなさい。また、何故このプログラムが例題 3 のプログラムより効率が良いのかを考えなさい (ヒント...前のプログラムでは m^8 を計算するのに $m^8 = m * m^7$, $m^7 = m * m^6$, ... としていたのに対し、今度のプログラムでは $m^8 = (m * m)^4$, $(m * m)^4 = ((m * m) * (m * m))^2$, ... としている)。

```
1 /* m の n 乗 (効率がいい解法) */
2 #include <stdio.h>
3
4 main ()
5 {
6     int m, n, result;
7
8     scanf("%d %d", &m, &n);
9     result = 1;
10    while (n > 0) {
11        while (n % 2 == 0) {                /* n が偶数の間、 */
12            n = n / 2;                      /* n を 2 で割った商を新たな n とし、 */
13            m = m * m;                      /* m * m を新たな m とすることを */
14        }                                    /* 繰り返す */
15        result = result * m;
16        n = n - 1;
17    }
18
19    printf("%d\n", result);
20 }
```

¹この繰り返しアルゴリズムは紀元前 200 年以前に Acharya Pingala によって書かれたヒンズーのチャンダースートラの中にある。