

## 2005年度 プログラミング演習 II 演習 11

2006年1月11日(水)

**演習 1** 以下は「ニュートン法を用いて2の平方根を求めるプログラム」である。このプログラムを読み取りなさい。また、このプログラムを変更し、「ニュートン法を用いて2の立方根を求めるプログラム」を作成しなさい(ファイル名「p2exercise1102.c」)。

● ニュートン法を用いて2の平方根を求めるプログラム

p2exercise1101.c

```
1: #include <stdio.h>
2: #include <stdlib.h> /* for exit() */
3:
4: #define MAX 30                                     ← 反復回数の制限
5:
6: double f(double x);
7: double df(double x);
8:
9: int main(void)
10: {
11:     int n = 1;
12:     double x = 2.0, y;
13:
14:     y = x - f(x) / df(x);
15:     printf("step%2d : %.15f\n", n, y);
16:     while (y != x) {
17:         x = y;
18:         y = x - f(x) / df(x);
19:         n++;
20:         printf("step%2d : %.15f\n", n, y);
21:         if (n > MAX) exit(1);                       ← 収束しない場合は強制終了
22:     }
23:     printf("2の平方根の近似値 : %.15f\n", y);
24:
25:     return 0;
26: }
27:
28: double f(double x)
29: {
30:     return x * x - 2;
31: }
32:
33: double df(double x)
34: {
35:     return 2 * x;
36: }
```

**演習 2** 以下は「モンテカルロ法を用いて円周率  $\pi$  の近似値を求めるプログラム」である。空欄  を埋めプログラムを完成しなさい。ヒント：■ p.273~p.274 を参考にすること (乱数の復習)。

● モンテカルロ法を用いて円周率  $\pi$  の近似値を求めるプログラム

p2exercise1103.c

```
1: #include <stdio.h>
2: #include <stdlib.h> /* for rand(), srand() */
3: #include <time.h> /* for time() */
4:
5: #define N 1000000                                     ← ばらまく点の個数
6:
7: int main(void)
8: {
9:     int i, n = 0;
10:    double x, y;
11:
12:     ;          ← 起動のたびに異なる乱数を発生させる設定
13:
14:    for (i = 1; i <= N; i++) {
15:        x =  ;          ← 0 から 1 未満の乱数
16:        y =  ;          ← 0 から 1 未満の乱数
17:        if (x * x + y * y <= 1) n++;          ← 円盤領域に含まれる点をカウント
18:    }
19:    printf("πの近似値 : %.15f\n",  ); ← キャスト演算子を使用
20:
21:    return 0;
22: }
```

\* 「■」は教科書「新訂 新 C 言語入門 ビギナー編」を表す。

**演習 3** 以下は「エラトステネスのふるいによって素数列を求めるプログラム」である。このプログラムを読み取りなさい。また、ヒントを参考に空欄  を埋めプログラムを完成しなさい。ヒント：

- 数の並びと配列の添え字が対応するように  $N+1$  個の配列を準備する ( $n \iff a[n]$ )。
- 2 ( $a[2]$ ) から  $N$  ( $a[N]$ ) までの数を扱う。
- 消されていない数には1を、消された数には0をセットする。
- 第12行には、「 $\lceil \sqrt{N} \rceil$  以下の素数の倍数を消せば十分である」という条件を入れる (数学関数 `floor()` を使用する)。
- 第13行では、 $N$  以下の素数  $p$  の倍数を全て消去 (数学関数 `floor()` を使用する)。

● エラトステネスのふるいによって素数列を求めるプログラム

p2exercise1104.c

```
1: #include <stdio.h>
2: #include <math.h> /* for floor(), sqrt() */
3:
4: #define N 30                ← 求める素数列の範囲 (2~N)
5:
6: int main(void)
7: {
8:     int a[N+1], i, p = 2;
9:
10:    for (i = 2; i <= N; i++) a[i] = 1;          ← 配列の初期化
11:
12:    while (p <= ) {
13:        for (i = 2; i <= ; i++) a[p * i] = 0;
14:        p++;
15:        while (a[p] == 0) p++;                ← pの次に大きい素数を見つける
16:    }
17:
18:    for (i = 2; i <= N; i++) {
19:        if (a[i] == 1) printf("%2d¥n", i);
20:    }
21:
22:    return 0;
23: }
```