

第7章 C言語 基礎編

7.1 基本的なプログラム1

地球の公転周期は 365.2422 日で、1 年を 365 日にとすると、何年かのうちにズレが生じます。そこで、4 年に 1 度閏年を設けてズレを調整しています。私達が使っているグレゴリオ歴は次の規則に従っています。

- (1) グレゴリオ歴は 1582 年 10 月から実施されたもの。
- (2) 西暦の年数が 4 で割り切れ、かつ 100 で割り切れない年は閏年。
- (3) 西暦の年数が 400 で割り切れる年も閏年とする。

問題 1 グレゴリオ歴に従って、年と月を入力することによって、対応するカレンダーを表示するプログラムを作成しなさい。

問題 2 プログラムを作成する上で、西暦 1 年 1 月 1 日は何曜日にすればよいか答えなさい。ただし、グレゴリオ歴は 1582 年 10 月から実施されたもので、それ以前のカレンダーは意味を持たない。

問題 3 グレゴリオ歴を用いると 1000 年で何日ズレが生じるか計算しなさい。

7.2 基本的なプログラム2

確率を用いて問題を解くことをモンテカルロ法といい、コンピュータが得意とする分野です。例えば、コイン投げやサイコロ投げは何度も実験を行うことによって確率に近い値を得ることができます。このような方法を用いて π の数値を求めてみましょう。 $C = \{x^2 + y^2 \leq 1, x \geq 0, y \geq 0\}$ と $S = \{(0, 0), (1, 0), (0, 1), (1, 1)\}$ を頂点とする正方形の面積比は $\frac{\pi}{4} : 1$ となります。0から1の範囲で (x, y) に対応する2つの一様乱数を発生させることによって点は領域 S に一様にばらまかれるので、 C に含まれた点の数を数えれば C と S の点の数の比によって π の値を求めることができます。

問題 1 モンテカルロ法を用いて π の値を求めなさい。

7.3 基本的なプログラム3

次のプログラムはユークリッドの互助法によって最大公約数を求めるプログラムです。次の問に答えなさい。

● ユークリッドの互助法によって最大公約数を求めるプログラム

7-1.c

```
1: #include <stdio.h>
2: main()
3: {
4:     int x=254,y=34,tmp;
5:     printf("gcd(%d,%d)=",x,y);
6:     while(y!=0){
7:         tmp=x%y;
8:         x=y;
9:         y=tmp;
10:    }
11:    printf("%d",x);
12: }
```

問題 1 プログラム 7-1.c について各行に解説を付けなさい。

問題 2 x, y の大小関係に関係なく解が得られるようにプログラム 7-1.c を直しなさい。

問題 3 実行中に x, y の値を入力できるようにプログラム 7-1.c を直しなさい。

問題 4 プログラム 7-1.c を利用して x, y の最小公倍数を求めるプログラムを作成しなさい。

問題 5 プログラム 7-2.c を参考に、プログラム 7-1.c をサブルーチンによる再帰的なプログラムに直しなさい。

● 再帰的に $n!$ を求めるプログラム

7-2.c

```
1: #include <stdio.h>
2:
3: int func(int x);
4:
5: main()
6: {
7:     int n=10;
8:     printf("%d!=%d",n,func(n));
9: }
10:
11: int func(int x)
12: {
13:     if(x==0) return 1;
14:     else return x*func(x-1);
15: }
```

7.4 基本的なプログラム 4

素数を列挙する簡単で高速なアルゴリズムとしてエラトステネス (Eratosthenes) のふるいがあります。例として、2 から 20 までに含まれる素数は、エラトステネスのふるいを使って次のように求められます。

Step 1 2 から 20 までを列挙します。

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20

Step 2 列挙した最初の数 2 の倍数を全て消します。

2, 3, 5, 7, 9, 11, 13, 15, 17, 19

Step 3 残った数の中で 2 の次に大きな数 3 の倍数を全て消します。

2, 3, 5, 7, 11, 13, 17, 19

Step 4 残った数の中で 3 の次に大きな数 5 の倍数を全て消します。

2, 3, 5, 7, 11, 13, 17, 19

このように、 $4.472 \leq \sqrt{20}$ まで素数の倍数を消去すると素数を列挙することができます。

問題 1 エラトステネスのふるいを使って N までの素数列を求めるとき、 $\sqrt{N}$ の倍数まで消去すれば十分である理由を述べなさい。

問題 2 エラトステネスのふるいを使って2から10000に含まれる素数の配列を求めなさい。ただし、使用する配列は

配列	$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	...
フラグ	1	1	1	1	1	1	...
実際の数	2	3	4	5	6	7	...

とし、素数の倍数であるときフラグ1を0に書き変えるものとする。

問題 3 問題2では、2の倍数が配列の半分を使用するためメモリの無駄が多い。次の配列を用いて、メモリの使用量を節約したプログラムに改良しなさい。ただし、他の条件は問題2に従うものとする。

配列	$a[0]$	$a[1]$	$a[2]$	$a[3]$	$a[4]$	$a[5]$	...
フラグ	1	1	1	1	1	1	...
実際の数	3	5	7	9	11	13	...

問題 4 問題3のプログラムを改良して、計算結果をファイルに出力するプログラムにしなさい。

問題 5 問題3で作成したプログラムをポインタを使ったプログラムに直しなさい。