



## 演習問題 7.1 文法事項

- 1) 暗黙の型変換とは何か答えなさい。

**解答例** 代入演算子(=)や算術演算子(+,-,\*,/,%)では、2つの演算項のデータ型が揃っている事が必要です。2つの演算項のデータ型が異なる場合、可能ならば、演算項のデータ型を変換しデータ型を揃える操作が暗黙のうちに実行されます。これを、暗黙の型変換と呼びます。

- 2) 暗黙の型変換のルールを2つ答えなさい。

**解答例** 1) 代入演算子が適用される際、代入される側のデータ型が、代入する先のデータ型へと型変換される。  
2) 算術演算子等の場合、2つの演算項のうち、より大きな型の方に合わせて、他方のデータ型が型変換される。

- 3) 明示的な型変換を行うための演算子の名称を答えなさい。また、その演算子の例をあげなさい。

**解答例** 明示的な方変換を行うための演算子をキャスト演算子と呼びます。キャスト演算子は、以下の例のように変換したいデータ型名をカッコで囲んだ形をしています。

(int), (double) など

- 4) プログラム例 7.1 中の剰余演算をうまく実行させる方法を考えなさい。

**解答例** そもそも、変数や定数のデータ型を全て int 型などの整数型に変更してしまえば、プログラム例 7.1 は動作します。何らかの理由で、変数のデータ型を int 型に出来ない場合は、下記のように、キャスト演算子によって、明示的に int 型に型変換すれば、演算が可能となります。

```
answer = (int)data % 3;
```

この例では、左辺で int 型の演算が行われた後、answer への代入時に、暗黙の型変換によって、演算結果が再び double 型に変換されることに注意してください。

- 5) 標準数学関数を用いるための注意事項を答えなさい。

**解答例** 標準数学関数を用いる場合には、プログラム中に、以下の命令を書き加え、math.h をインクルードする必要があります。

```
#include <math.h>
```

また、コンパイラ(正確にはリンカ)に対して、標準数学関数のライブラリを利用することを指示する必要があります。例えば、gcc コンパイラ等では、-lm スイッチを付けて下記のようにコンパイルします。

```
cc -o prg -lm prg.c
```

## 演習問題 7.2

極座標  $(r, \theta)$  から直交座標  $(x, y)$  を求めるプログラムを書きなさい。

**解答例** 解答例を解答プログラム 7.1 に示します。

標準数学関数のおかげで、ほとんど数学の式そのままにプログラムを書くことができるのがわかると思います。有り難いことです。

### 解答プログラム 7.1

### 座標変換プログラム

```

/*****
p7_1.c
座標変換プログラム
Tadaaki Shimizu    2006.11.15
*****/
#include <stdio.h>
#include <math.h>

main()
{
    double radius, theta;    // 半径, 偏角
    double xCoord, yCoord;   // x座標, y座標

    // 極座標入力
    printf(" 極座標 : 半径 > ");
    scanf("%lf", &radius);
    printf(" 極座標 : 偏角 > ");
    scanf("%lf", &theta);

    // 座標変換と出力
    xCoord = radius * cos(theta);
    yCoord = radius * sin(theta);
    printf("(x, y) = (%lf, %lf)\n", xCoord, yCoord);

    exit(0);
}

```

## 演習問題 7.3

プログラム例 7.5 を参考に、指数関数と自然対数関数の表を適当な範囲で作成するプログラムを作りなさい。

**解答例** 解答例を解答プログラム 7.2 に示します。

数表を作る範囲と刻みは、`#define` 命令で定義した、`START`、`END`、`DELTA` で設定しています。

プログラムの実行で、「毎回、設定値を変えたい」という場合には設定値の入力を受け付けるようにプログラムを設計します。しかし、「もしかしたら将来、設定値を変えるかも」という程度であれば、設定値の入力を毎回要求されるようなプログラムはかえって使いにくいものです。

このような場合、解答プログラム 7.2 のように定数定義を使って設定値を定義しておくと、設定値を変えたいときには、プログラム全体に目を通す必要は無く、定

数定義の部分だけを変更すればよいので非常に便利です。

解答プログラム 7.2 をそのまま実行すると、`arg` が `0.0` のときに、`log` の方の答えが `-inf` と表示されるはずですが、(IEEE 方式による) 浮動小数点数の計算では、計算結果が負の無限大で `-inf`、正の無限大で `inf`、正しい数を計算できない場合 (非数) は、`NaN` という名前が付いた特別な値になります。これらは、数として使えない値ですが、`printf()` 関数などで表示すると、アルファベットでそれぞれの名前が表示されます。デバッグの際などに重宝しますので、覚えておくとよいでしょう。

## 解答プログラム 7.2

## 指数／対数表を作るプログラム

```
/* *****  
p7_2.c  
exp, log の表を作る  
Tadaaki Shimizu      2006.11.15  
***** */  
#include <stdio.h>  
#include <math.h>  
#define START 0.0      // 表の開始  
#define END 3.0        // 表の終了  
#define DELTA 0.1      // 表の刻み  
  
main()  
{  
    double arg;          // 関数の引数  
    double exponential;  // exp の答え  
    double logarithm;    // log の答え  
  
    // START から END まで DELTA 刻みで表を作る  
    printf("arg      exp      log\n");  
    for(arg = START; arg <= END; arg = arg + DELTA) {  
        exponential = exp(arg);  
        logarithm    = log(arg);  
  
        printf("%lf : %lf , %lf\n", arg, exponential, logarithm);  
    }  
    exit(0);  
}
```

### 演習問題 7.4

2次方程式の係数を入力すると、その解を求めるプログラムを作りなさい。解の種類(2実根, 重根, 複素根)のどれにも対応できるようにプログラムしなさい。

**解答例** 解答例を解答プログラム 7.3 に示します。

#### 解答プログラム 7.3

#### 2 次方程式の解を求めるプログラム

```

/*****
p7_3.c
2 次方程式の解
Tadaaki Shimizu      2006.11.15
*****/
#include <stdio.h>
#include <math.h>

main()
{
    double aCoef, bCoef, cCoef; // 2 次方程式の係数
    double disc;                // 判別式
    double aPart, bPart;        // 解の前半分と後半分

    // 2 次方程式の係数入力
    printf("2 次方程式の係数\n");
    printf(" a = ");
    scanf("%lf", &aCoef);
    if(aCoef == 0.0) {
        printf("ERROR: a の値が 0.0 です.\n");
        exit(0);
    }
    printf(" b = ");
    scanf("%lf", &bCoef);
    printf(" c = ");
    scanf("%lf", &cCoef);

    // 2 次方程式の求解
    disc = bCoef * bCoef - 4.0 * aCoef * cCoef;
    if(disc > 0.0) {
        aPart = - bCoef / (2.0 * aCoef);
        bPart = sqrt(disc) / (2.0 * aCoef);
        printf("2 実根 : %lf, %lf\n", aPart + bPart, aPart - bPart);
    }
    else if(disc == 0.0) {
        aPart = - bCoef / (2.0 * aCoef);
        printf(" 重根 : %lf\n", aPart);
    }
    else {
        aPart = - bCoef / (2.0 * aCoef);
        bPart = sqrt(-disc) / (2.0 * aCoef);
        printf(" 複素根 : %lf + %lf i\n", aPart, bPart);
        printf("          %lf + %lf i\n", aPart, bPart);
    }
    exit(0);
}

```

## 演習問題 7.5

あるガソリンスタンドでは、ガソリンを20リットル以上入れた人には5%引きで、40リットル以上入れた人には10%引きで売ります。その日のガソリンの1リットルあたりの価格と、入れた量を入力して支払額を計算するプログラムを作りなさい。ただし、支払額は小数点以下まで表示すること。

**解答例** 解答例を解答プログラム 7.4 に示します。

### 解答プログラム 7.4

### ガソリン支払額計算プログラム

```
/*
*****
p7_4.c
ガソリンの価格
Tadaaki Shimizu      2006.11.15
*****
#include <stdio.h>

main()
{
    double price;    // ガソリンの価格
    double volume;   // ガソリンの量
    double payout;   // 支払額

    // データ入力
    printf(" ガソリンの価格 (円/L) : ");
    scanf("%lf", &price);
    if(price < 0) {
        printf("ERROR : 価格が負の値です.\n");
        exit(0);
    }
    printf(" ガソリンの購入量 (L) : ");
    scanf("%lf", &volume);
    if(volume < 0) {
        printf("ERROR : 購入量が負の値です.\n");
        exit(0);
    }

    // 支払額の計算
    if(volume >= 40) {
        payout = price * volume * 0.9;
    }
    else if(volume >= 20) {
        payout = price * volume * 0.95;
    }
    else {
        payout = price * volume;
    }
    printf("%lf(L) お買い上げ %lf 円です.\n", volume, payout);
    exit(0);
}
```