情報処理 第11回
Cで計算しよう

2000年7月5日

目 次

1 本日の目標 **1**

2 前回の復習 **1**

3 基本データ型 **2**

4 四則演算 **4**

5 ユークリッドの互除法 **5**

6 お知らせ **5**

1 本日の目標

- ユークリッドの互除法を利用したソースを書く.
- While 文を理解する.

2 前回の復習

- C 言語のソースは

```
#include<stdio.h>

main()
{
     (本文)

 }
```

が基本であり, ソースファイルは

(ファイル名).c

と拡張端子.c を付け忘れないこと. また, コンパイルするときは

cc (ファイル名).c

とする. そして, 起動させるときは a.out とする.

- for 文は次のような形で使用します .

```
for (式 1 ; 式 2 ; 式 3 ) {
    … ;
}
```

式 1 は , for 文の中の処理を行う前に 1 度だけ行います . 次に式 2 を判定し , その結果が真ならば , {} で囲まれた部分を実行します . その後 , 式 3 を実行し , 再び式 2 を判定します . あとはこの繰り返しです .

例として先週のレポート問題 creportno.10-2 解答を載せます.

```
#include<stdio.h>

main( )

{

    int     i;
    int     sum;
```

```
    sum = 0;


  for (i = 0; i < 100; i++) {

    sum = sum + 3*i+11 ;
  }


  printf("sum = %d\n", sum);
}
```

3 基本データ型

C を用いたプログラミングでは，変数を使うときには必ずそのデータの型を指定して，その名前を宣言する必要があります．

変数の名前は，アルファベットで始まる 31 文字以下の文字の並びです．(31 文字を超える変数名を使うと，32 文字目以降は無視される．) 実際にプログラムを動かすときデータは，主記憶に格納されます．機械語のレベルでは，この時に必要な記憶容量を正確に知っておく必要があります．つまり Compiler は，変数の宣言によって必要となる記憶容量を計算し，これを元にして，プログラムの内容を機械語に翻訳していくのです．この記憶要領の計算の元になるのが，データの型と言われるものです．

C 言語におけるもっとも基本的なデータの型は，以下のとおりです．これ以外にも，データの型というものがあるのですが，それらについては後の講義でふれます．

char 1 文字を表す．通常は， 1 byte = 8 bit．皆さんの使っている処理系もそうです．ただ，これでは日本語の漢字等は表示できないので，漢字表示が必要な場合には特別なことをします．この講義ではその事には触れません．

int 整数型．これは数学的な無限の整数を扱える訳ではなく，有限の範囲の整数しか扱えません．扱える範囲は処理系によって様々です．RAINBOW のマシンでは int のサイズは 4 byte = 32 bit です．

float 浮動小数点型．実数を扱うときに使用します．現在ではあまり使いません．RAINBOW のマシンでは float のサイズは 4 byte = 32 bit です．

double 倍精度浮動小数点型．float では計算の誤差が多く，精度が確保できないときに使うために，設けられたのが起源です．しかし現在では，この double を使うのが普通です．RAINBOW のマシンでは double のサイズは 8 byte = 64 bit です．

例えば 2^n を求めるソースファイルを考えてみましょう.
ファイル名を 11-1.c として書いてみてください.

```
#include <stdio.h>

main()
{
int i, n; /*変数が整数型であることを宣言*/
 int prod; /*同じく型を宣言*/
 printf("2^n を求めます。>>\n");
 printf("指数 n を代入してください。>>\n");
 scanf("%d",&n); /*データを入力*/

 prod = 1; /*変数を初期化します*/

 for(i = 0;i < n;i++){
     prod = prod * 2;
   }
  printf("2^n = %d です\n",prod);

}
```

このプログラムでは,「指数 n を代入してください。>>」をディスプレイ上に書き出し, キーボードから打ち込まれた自然数 n を記憶して 2^n をだすようにしています. このようなプログラムをインタラクティブなプログラムといいます. インタラクティブ (interactive) とは , 会話的という意味です .
関数 scanf の使い方は,

```
scanf("%d",&a)
```

で, int 型の変数 a にデータを入れます. ただし、double 型の変数 x の場合は,

```
scanf("%lf",&x)
```

となります. "l"(エル) を忘れずに!!.
変数 a の前の記号&は、アンパサンド といい, 変数 a のアド レスを与えています. 今は, おまじないと思って下さい.
11-1.c を書き換えたあとコンパイルをし起動させてみてください. $n = 31$ を入れたらどうなりますか？

次に整数の四則演算について考えてみます.

4　四則演算

次のプログラムを 11-2.c という名前で作成して下さい．出来たら，コンパイルして実行してみて下さい．

```
#include <stdio.h>

main()
{
     int     a, b;        /* 変数 a,b が整数型であることを宣言 */
    int     sum, diff, prod, quot, rmind;  /* 同じく型宣言 */

    printf("二つの整数 a,b の四則計算をします.>>\n");
    printf("a, b に代入する数を空欄をいれて連続して入力して下さい\n");
    scanf("%d %d", &a, &b);          /* a,b にデータ入力 */



    sum = a + b;         /* sum に a+b を代入します */
    diff = a - b;        /* diff に a-b を代入します */
    prod = a * b;        /* 掛け算は * という記号を使います */
    quot = a / b;        /* ひき算は / という記号を使います */
    rmind=a % b;          /* これは何でしょう? */

    /* printf は"で囲まれた書式に従って出力します */
    printf("summation = %d, difference = %d\n", sum, diff);
    printf("product = %d, quotient = %d, reminder = %d\n", prod, quot,rmind);
}
```

これもインタラクティブなプログラムです．さて

```
rmind = a % b
```

は何でしょうか？例えば，$a = 3, b = 2$, $a = 4, b = 2$ を代入して考えてみてください．

5　ユークリッドの互除法

ユークリッドの互除法とは次のことを意味します.

定義 $a, b(b \neq 0)$ を正の整数としたとき, ある整数 q と r が存在し,

$$a = bq + r$$

とかける. 但し, $0 \leq r < b$ ととれる. もし, $r = 0$ ととれれば, a, b の最大公約数は b であることがわかる.

考察

今, 与えられた正の整数 a, b に対して, $r = 0$ となるまで繰り返しをする. 即ち,

一回目 $a = bq_1 + r_1$
2 回目 $b = r_1 q_2 + r_2$
3 回目 $r_1 = r_2 q_3 + r_3$
　　...
n 回目 $r_{n-2} = r_{n-1} q_{n-2}$

このとき, r_{n-1} が a, b の最大公約数.　(何故か？)

これを用いて, キーボートから入力した正の整数の最大公約数を求めるプログラムを書きましょう. できた方は, ereportno.11-3 とし osaka-report まで提出してください. 時間内にできない方は来週 7 月 11 日までに提出すること. 2 つの正の整数 a, b を入力させるのだから, 関数 scanf は

```
scanf("%d %d", &a,&b)
```

としなければならない.

また, 繰り返しのアルゴリズムであるので,　for 文を用いることになるが, 次の while 文の方がきれいに書けます.

while 文

while 文は次のように使います .

```
while (条件式) {
  条件式が真ならここを実行
}
```

上の場合, 条件式は余り r がゼロでない限り, ユークリッドの互除法を繰り返せということですから, 何を入れたらよろしいかわかると思います.

6　お知らせ

出席, レポートの評価などについては下記の URL でチェックしてください:

```
http://www.ritsumei.ac.jp/~osaka/rejime/jhou/seiseki.html
```