



Title	計算機プログラミングI・同演習 講義ノート2007
Author(s)	井上, 純一
Description	2007年度前期に開講された工学部情報エレクトロニクス学科2年生を対象としたLinuxシステム、C言語プログラミングに関する入門的な講義・演習の講義ノートです。この講義・演習で扱わない、より進んだ内容は後期に開講される「計算機プログラミングII」にて学習します。
Issue Date	2007-08-22T04:23:05Z
Doc URL	https://hdl.handle.net/2115/28047
Rights(URL)	https://creativecommons.org/licenses/by-nc-sa/2.1/jp/
Type	learning object
File Information	ProgI2007_4.pdf, 第4回講義・演習ノート



計算機プログラミングI 講義ノート #4

担当：井上 純一 (情報科学研究科棟 8-13), 赤間 清 (情報基盤センター)

URL : http://chaosweb.complex.eng.hokudai.ac.jp/~j_inoue/PROG2007/PROG2007.html

平成 19 年 4 月 27 日

目次

11.3 switch case 文	30
12 数学関数	32
13 反復制御	33
13.1 while 文	33
13.2 for 文	34

11.3 switch case 文

流れの処理を行わせるには, 先週学んだ if 文, if else 文の他に switch case 文があります. 今回はまずこの使い方について詳しくみていきましょう.

switch case 文の使い方は以下の通りです.

```
switch(式){
    case 定数 1: 文 1.....
        break;
    case 定数 2: 文 2.....
        break;
    case 定数 3: 文 3.....
        break;
    .....
    .....
    .....
    case 定数 n: 文 n.....
        break;
    default: 文 n+1.....
        break;
}
```

このとき, 処理の流れはまず, (式) が評価され, その値と各 case の後に書かれた定数を順次に比較します.

その再、もし、(式) と一致する定数 k が存在すれば、その後の文 k 以降を実行されます。その実行中に `break;` に出会うと、`switch case` 文全体の実行を終了します。もし、`break;` が無ければ `switch` 文の終わりまでの文を順次に行うこととなります。この `switch case` 文の使い方に慣れるために次の **練習問題 11.1** をやってみましょう。

練習問題 11.1

次のプログラムを作成、コンパイルし、実行せよ。特に、`break` 文の位置による違いを確認せよ。

```
#include<stdio.h>
main()
{
    int x;
    printf("x=");
    scanf("%d",&x);

    printf("***** break in all three lines *****\n");
    switch(x)
    {
        case 1: printf("1\n"); break;
        case 2: printf("2\n"); break;
        case 3: printf("3\n"); break;
    }

    printf("***** no break in all three lines *****\n");
    switch(x)
    {
        case 1: printf("1\n");
        case 2: printf("2\n");
        case 3: printf("3\n");
    }

    printf("***** break in the second line *****\n");
    switch(x)
    {
        case 1: printf("1\n");
        case 2: printf("2\n"); break;
        case 3: printf("3\n");
    }
}
```

この練習問題での結果を念頭に次の問題を LMS 入力問題として考えてみてください。

練習問題 11.2 (LMS 入力問題)

各種運動を一定時間行った場合の消費カロリーが次表で与えられている。

運動種目番号	運動種目	消費カロリー (K cal/分)
1	縄跳び	8.60
2	ジョギング	6.25
3	エアロビクス	4.41
4	散歩	2.25

このとき、運動種目番号と運動時間(分)をキーボード入力から与えると消費カロリーを計算して画面表示するプログラムを作成せよ。

「運動種目番号」を switch case 文の case の番号に対応させ、分岐を行わせませよ。

12 数学関数

科学計算、工学計算では $\sin x$, $\cos x$ などの数学関数を用いるのが普通です。C 言語には予め用意されている数学関数があり、これは math.h というファイルの中で定義されています。これらの数学関数を用いるためには、プログラムに

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

という一行を書き込んでおく必要があります。また、既に注意を促していましたが、コンパイルを実行する際には

```
gcc -o b.out test.c -lm
```

のように `-lm` オプションをつけることを忘れないようにしてください。

以下に主だった数学関数を列記しておきます。

関数	説明
$\sin(x)$	x の sine(正弦), 単位はラジアン
$\cos(x)$	x の cosine(余弦), 単位はラジアン
$\tan(x)$	x の tangent(正接), 単位はラジアン
$\text{asin}(x)$	$\sin^{-1}(x)$, x の範囲 $[-1, 1]$, 関数値の範囲 $[-\pi/2, \pi/2]$
$\text{acos}(x)$	$\cos^{-1}(x)$, x の範囲 $[-1, 1]$, 関数値の範囲 $[-\pi/2, \pi/2]$
$\text{atan}(x)$	$\tan^{-1}(x)$, 関数値の範囲 $[-\pi/2, \pi/2]$
$\text{atan2}(x,y)$	$\tan^{-1}(y/x)$, 関数値の範囲 $[-\pi, \pi]$
$\log(x)$	x の自然対数 $\ln(x)$, ($x > 0$)
$\log10(x)$	x の常用対数 $\log(x)$, ($x > 0$)
$\text{pow}(x,y)$	x の y 乗. x^y
$\text{fabs}(x)$	x の絶対値
$\text{sqrt}(x)$	x の平方根. \sqrt{x} ($x > 0$)
$\text{floor}(x)$	x より大きくない最小の整数
$\text{ceil}(x)$	x より小さくない最小の整数

このリストにない関数でも、後に見るように、自分で必要な関数を作成し、それをメイン関数の中に呼び込んで用いることもできます。次の練習問題をやってみましょう。

練習問題 12.1 (LMS 入力問題)

三角形の 3 つの辺の長さ a, b, c をキーボードより入力すると, その面積 s を次の公式:

$$s = \sqrt{t(t-a)(t-b)(t-c)}, \quad t = (a+b+c)/2$$

によって計算し,

s=****

の形式で画面に表示させるプログラムを作成せよ. ここで, a, b, c が三角形をなすためにはどの辺も他の 2 辺の和よりも小さくなくてはならないことに注意し (三角不等式), もし, この条件を満たさない場合には impossible と表示するようにプログラムを書くこと.

13 反復制御

ここからは, ある作業を計算機に繰り返し行わせるための方法 — ループ処理 (繰り返し処理) — のいくつかを学んでいきます.

13.1 while 文

ループ処理の中でも while 文と呼ばれる反復制御法は次のような構造を持っています.

```
while(条件式){
    文 1;
    文 2;
    .....
    .....
    文 n;
}
```

この while 文は「条件式」の値が 0(偽) になるまで次の手順でループ本体を繰り返します.

- (1) 繰り返しの判定: 「条件式」を評価し, s の値が 0(偽) ならば, この while 文を終了し, {...} に続く文の実行に移る.
- (2) ループ本体の実行: ループ本体を実行し, (1) 戻る.

具体的な使い方を見るために, 次のような練習問題を見ておきましょう.

練習問題 13.1

次のプログラムをファイルに打ち込み, コンパイル・実行し, while 文の動作を確認せよ.

```
/* while 文の動作の確認 */
/* s=1,...,10 に対し, それぞれの逆数, 2 乗, 平方根を表示させる */
#include<stdio.h>
#include<math.h>
```

```
main()
{
    int s;
    double a,b,c,value;
    s = 1;
    printf("Number Inverse Square Root\n");
    printf("-----\n");

    while(s <= 10)
    {
        value = s;
        a = 1/value;
        b = pow(value,2);
        c = sqrt(value);

        printf("%f %f %f %f\n", value, a,b,c);
        s = s + 1;
    }
}
```

いくつかの注意事項

- (1) while 文の処理がただ一つの文だけの場合には中括弧 {} を省略できる.
- (2) 次のような while 文を無限ループと呼ぶ.

```
/* 無限ループの例 */
while(1)
{
    printf("beautiful life!\n");
}
```

プログラミングの際にこのような無限ループを作ってしまうと、永遠にジョブが終了しないので、絶対に避けなければならない。自分でも気づかないうちに無限ループを作っている場合もあるので、なかなかジョブが終了しない場合には Ctrl-C でジョブを強制的に終了し、無限ループができていないかどうかをチェックすると良い。

13.2 for 文

繰り返し処理を行わせるには次に挙げる for 文を使うこともできます。

```
for(式 1; 式 2; 式 3)
{
    文 1;
    文 2;
    .....
    .....
    文 n;
}
```

for 文では、式 2 の値が 0(偽) になるまで、次のようにループ本体が繰り返し実行されます。

- (1) ループの初期化: 式 1 を評価し、繰り返し用いる変数に必要な初期値を代入する。
- (2) 繰り返しの判定: 式 2 を評価し、その値が 0(偽) ならば for 文の実行を終了し、そうでなければ次のステップへ移る。
- (3) ループ本体の実行: ループ本体の文を実行する。
- (4) 繰り返し後の演算: 式 3 を評価する。式 3 は繰り返しを制御する変数の値を増加、あるいは減少させる。その後 (2) へ移る。

ここでも for 文の使い方を実際に見てもらうため、次のような練習問題をやってみましょう。

練習問題 12.2

次のプログラムをファイルに書き込み、コンパイル・実行し、for 文の動作を確かめよ。

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

main()
{
    int i,n;
    n=0;
    for(i=1; i<=10; i=i+1){
        n=n+i;
        printf("sum is %d\n",n);
    }
}
```

注意

for 文の場合にも無限ループができるので注意が必要です。次のように書くと無限ループができます。

```
for( ; ; ){
    printf("beautiful life!\n");
}
```

今週の LMS 入力問題

(1) $s = \sum_{k=1}^{10000} (6/k^2)$ を計算し

s=*****

の形式で表示するプログラムを作成せよ。

(2) 漸化式:

$$x_{n+1} = \frac{1}{2} \left(x_n + \frac{s}{x_n} \right), \quad x_1 = 3$$

を考えよう。この漸化式の $n = 100$ の項を

x=*****

の形式で表示させるプログラムを作成せよ。ただし、この漸化式の中に現れる s の値は (1) で求めたものであるとする。

参考 1: C 言語において $a=b$ とは b の値を a に「代入する」ことであったことを思い出すと良い。

参考 2: どのようにプログラムを組んだらよいのかは、具体的に漸化式のはじめの数項を紙に書き出して「感じ」をつかむと良い。「面倒だな」と思った部分を計算機に渡して処理させる。

(3) (2) での漸化式が収束する場合にその収束値を求めたい。今回学習した for 文を用いて繰り返しの最大値を 10000 程度とし、条件: $|x_{n+1} - x_n| < 10^{-5}$ を満たすまで反復を繰り返し、この条件が満たされた時点での x_n の値を

x=*****

の形式で表示させるプログラムを作成せよ。このとき、うまくプログラムが動作していれば、この収束値は円周率 π の良い近似値になっているはずである。

注: 今週の問題の LMS への入力期限は 5/10(月) 午後 5 時までとします。