

# 物理学情報処理演習

## 6. C言語②

2015年5月22日

本日の推奨作業directory

lesson06

6.1 演算(算術以外)

6.2 制御文

6.3 グラフ表示ソフトgnuplot

参考文献

- やさしいC++ 第4版 高橋 麻奈 (著)

ソフトバンククリエイティブ

- プログラミング言語C++第4版

ビャーネ・ストラウストラップ, Bjarne Stroustrup, 柴田 望洋

- Numerical Recipes: The Art of Scientific Computing, Third Edition in C++

身内賢太朗

レポート提出: [fsci-phys-jouhou@edu.kobe-u.ac.jp](mailto:fsci-phys-jouhou@edu.kobe-u.ac.jp)

# 6.1 演算

## 6.1.1 演算 (算術以外)

- 代入演算子 =
- 関係演算子 == (等しい), > (より大きい), < (より小さい), >= (以上), <= (以下)
- 論理演算子 && (且つ), || (または), != (等しくない)
- ビット演算子 &, |, !, ^

# 6.2 制御文

## 6.2.1 if 文

○条件の真偽で判定

```

if(条件){
    実行文(条件が真のとき);
}
else{
    実行文(条件が偽のとき);
}

```

```

#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
    //calculate the area of a triangle.
    //command line parameter is angle C(degree).
    int i=1;
    double a,b;
    double ans,myans;
    char message[128];
    a=2.;
    b=10.;
    ans=pow(a,b);
    strcpy(message,"What is 2 to the 10th?(Calculate 2^{10})>");
    cout <<message;
    cin>>myans;
    if(myans==ans){
        cout << "Right !" <<endl;
    }
    else{
        cout << "No..." <<endl;
    }

    cout << "Your answer is " << myans <<endl;
    cout << "Real answer is " << ans <<endl;
    return 0;
}

```

## 6.2.1 if 文(続)

○条件分岐が3つ以上のとき、else if を使って

```
if (条件1){
    実行文1(条件1が真のとき);
}
else if (条件2){
    実行文2(条件2が真のとき);
}
else {
    実行文3(条件1,2 が共に偽のとき);
}
```

$a < b$ かつ $b < c$  という条件を課す際  
 $a < b < c$  としてはいけない。

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
    //calculate the area of a triangle.
    //command line parameter is angle C(degree).
    int i=1;
    double a,b;
    double ans,myans_l,myans_u;
    char message[128];
    a=2.;
    b=10.;
    ans=pow(a,b);
    strcpy(message,"Estimate 2 to the 10th ?(Calculate 2^{10})");
    cout <<message<<endl;
    cout <<"lower?>> ";
    cin>>myans_l;
    cout <<"upper?>> ";
    cin>>myans_u;
    if(myans_l>myans_u){
        cout << "The lower limit is larger than the upper limit. Exiting" <<endl;
        return 1;
    }
    if(myans_u==ans||myans_l==ans){
        cout << "Bingo!" <<endl;
    }
    else if(myans_l<ans&&ans<myans_u){
        //never write myans_l<ans<myans_u !!
        cout << "Your estimation is right. The answer in in your range." <<endl;
    }
    else{
        //never write myans_l<ans<myans_u !!
        cout << "Your estimation is wrong. The answer in out of your range." <<endl;
    }
    return 0;
}
```

power\_2.cxx

power\_2.cxx :2の10乗を推定するプログラム。

## 6.2.2 switch 文

○switchによる分岐も可能

```
switch (変数) {
    case 値1:
        実行文1;
        break;
    case 値2:
        実行文2;
        break;
    default:
        実行文3
        break;
}
```

caseの値として取れるのは、整数もしくは文字

演習6.2 次のようなプログラムを書いてみよう(提出は不要)

ある実数の平方根を推定するプログラム。

実数はコマンドライン引数として、上限と下限をプログラム中で入力する。

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;
int main(int argc, char *argv[]){
    //calculate the area of a triangle.
    //command line parameter is angle C(degree).
    int i=1;
    double a,b;
    double ans,myans;
    char message[128];
    a=2.;
    b=10.;
    ans=(int)pow(a,b);
    strcpy(message,"What is 2 to the 10th?(Calculate 2^{10})");
    cout <<message;
    cin>>myans;
    switch((int)(myans-ans)){
    case 0:
        cout << "Right!" <<endl;
        break;
    default:
        cout << "No..." <<endl;
        break;
    }
    cout << "Your answer is " << myans <<endl;
    cout << "Real answer is " << ans <<endl;
    return 0;
}
```

power\_3.cxx

power3.cxx : power\_1.cxxをswitch分を使って書き換えたもの。

## 6.2.3 繰り返し文

○For文: 決まった回数繰り返す

```
for (初期化; 条件; 終端処理){
    実行文;
}
```

○while文 条件が満たされる間繰り返す

```
while (条件) {
    実行文;
}
```

$i=i+1$ ; と  $i++$ ;  
は同じ意味  
 $i=i-1$ ; と  $i--$ ;  
は同じ意味

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;

int main(int argc, char *argv[])
{
    //calculated the factorial
    int n,i,ans;
    if(argc>1){
        n=(int)atof(argv[1]);
    }
    else{
        n = 1;
    }
    ans=1;
    for(i=1;i<=n;i++){
        //i++ is same as i=i+1
        ans=ans*i;
        cout << i << "¥t" << ans << endl;
    }
    cout << "# <<n<< != <<ans << endl;
    return 0;
}
```

fact\_1.cxx

¥tはtabで列区切りの記号

```
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
using namespace std;

int main(int argc, char *argv[])
{
    //calculated the factorial
    int n,i,ans;
    if(argc>1){
        n=(int)atof(argv[1]);
    }
    else{
        n = 1;
    }
    i=1;
    ans=i;
    while(i<=n){
        ans=ans*i;
        cout << i << "¥t" << ans << endl;
        i++;
        //i++ is same as i=i+1
    }
    cout << "# <<n<< != <<ans << endl;
    return 0;
}
```

fact\_2.cxx

fact\_2.cxx : 階乗を計算するプログラム。  
(要引数)

fact\_1.cxx : 階乗を計算するプログラム。(要引数)

**a=a+b;**

**a+=b;**

は同じ意味。

**+**の代わりに**-**, **/**, **\*** も使用可能

演習6.2.3 次のようなプログラムを書いてみよう(提出は不要)

fact\_1 5 の出力を fact\_5.datとしてセーブしておこう。(catで中身確認)

fact\_1を nから降順に掛け算してゆくものに変更する。 i-- を使う。 fact\_3.cxx

fact\_1を \*= を用いて書き換える。 fact\_4.cxx

<発展>

fact\_3を基にして、1行目に1から10までの整数 2列目にその階乗を表すプログラムを書いてみよう。 fact\_5.cxx

※for文を2重で使う必要あり。

## 6.3 gnuplot

cで計算した結果を簡単にグラフ化して確認したい

gnuplot : 簡単なグラフ描画ソフト

\$gnuplotで立ち上がる。以下、fact\_5.datのあるdirectoryでgnuplotを立てたとして例示。

```
gnuplot > plot "fact_5.dat"
```

で右の様に 1列目を横軸2列目を縦軸としたグラフが描ける。

```
gnuplot > set log y
```

← 縦軸をlogにする

```
gnuplot > setgrid
```

←gridを表示

```
gnuplot > plot [0:6][0.1:] 'fact_5.dat' using 1:2 title  
"factorial" with points 3
```

で右の様に 書ける[:]内はx,y,のrange

usingは列を指定 titleで凡例に付ける文字を指定

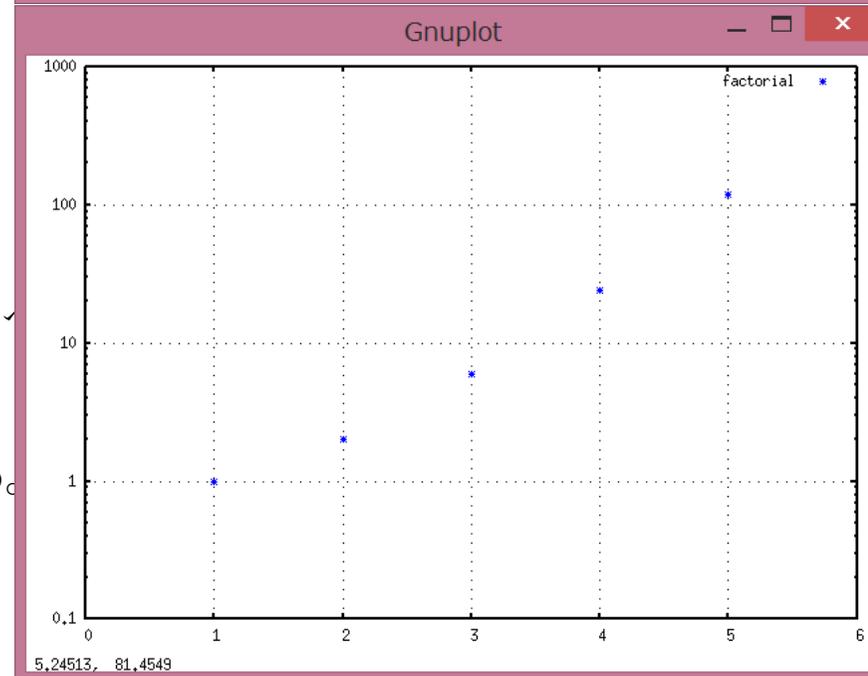
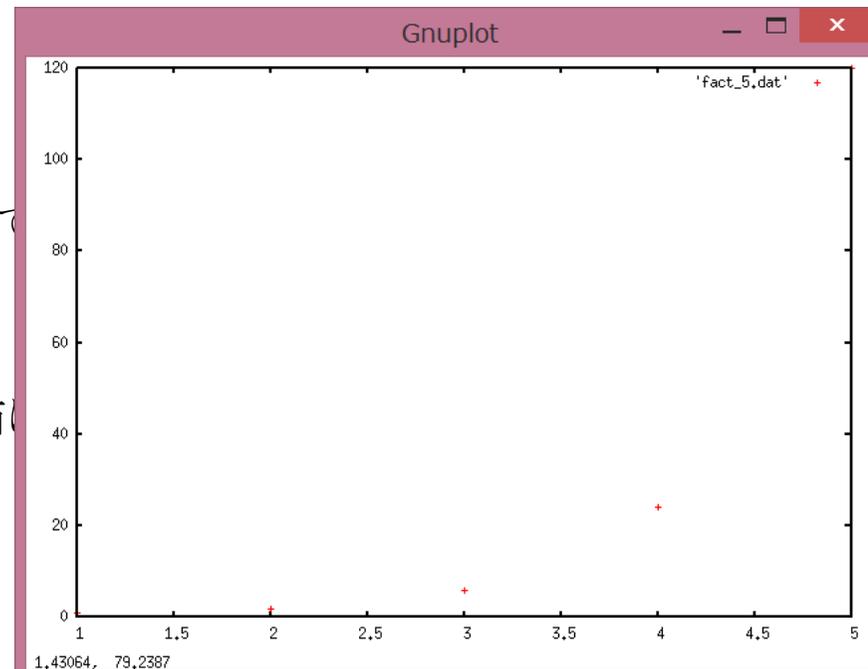
with 以下でマーカーの種類を指定。

マクロファイルと呼ばれるテキストファイルにコマンドを並べて記述して(show\_fact.pltをダウンロード)

```
gnuplot> load "show_fact.plt"
```

としても同じことができる。

上部のツールバーから画像をpdf fileとして保存可能。



## <gnuplotのコマンド例>

☆☆☆ plot “*filename*” *filename*をplot

☆☆☆ plot 式(x) 式をplot 式の例 x, x\*\*2, exp(x)

☆☆☆ replot “*filename*” *filename*を再描画

☆☆☆ load “*macrofilename*” *fmacrofilename*を読み込む

☆☆ (un)set log x(y) x(y) 軸をlogスケールにする(解除する)

☆☆ plotのオプション □

ソースコードファイル名:2015\_jouhou\_06\_学籍番号の下4桁.cxx  
出力ファイル名:2015\_jouhou\_06\_学籍番号の下4桁.dat  
出力画像ファイル名:2015\_jouhou\_06\_学籍番号の下4桁.pdf

課題6:以下の仕様のプログラムを製作し、 $\mu = 1.0$ のポアソン分布を計算せよ。ソースコード、出力ファイル、gnuplotで描画した出力結果の図を提出せよ。(for文を2重で使う必要あり。)

- ① 引数でポアソン分布の期待値  $\mu$  を取り込む
- ② 出力1列目:0から10までの整数
- ③ 出力2列目:期待値  $\mu$  ポアソン分布で実現値Xを得る確率

$$P(\mu, x) = \frac{\mu^x}{x!} e^{-\mu}$$

表示

横軸:リニアスケール (範囲は0~10)

縦軸:リニアスケール (範囲は0~0.5)