UEC The University of Electro-Communications

プログラミング演習

III類 クラスA
（機械システム/電子工学/光工学/化学生命工学プログラム）

選択必修/選択 2単位

〔第2回　数学関数ライブラリ〕

前回の演習課題

演習課題1-1

学籍番号をキーボードから入力すると類と学年を表示するプログラムを作りなさい。学域昼間コース1・2年生以外の学籍番号が入力された場合はその旨を表示すればよい。

コーディング例

```
 1  /*****************************************
 2     Student IＤ No. --> Cluster & Year
 3     Oct.10,2024  prog@ds.mce.uec.ac.jp
 4   *****************************************/
 5
 6  #include <stdio.h>
 7
 8  int main(void)
 9  {
10      int  id, year, course, cluster;
```

※変数名はa,b,c,dなどにせず内容が想像できるものにしたほうがよいです。

```
11
12      printf("学籍番号 >");
13      scanf("%d", &id);
14
15      year    = id/100000 ;
16      course  = (id/10000) % 10 ;
17      cluster = (id/1000)  % 10 ;
18
19      if( year<23 ) {
20          printf("1・2年生以外の学籍番号です\n");
21          return( 1 );
22      }
23      if( course != 1 ) {
24          printf("学域昼間コースの学籍番号ではありません\n");
25          return( 2 );
```

※適当なプロンプトを出したほうが使い勝手の良いことが多いです。

※学籍番号を文字列で処理しても構いません。ただし，配列は8要素必要です。

※%演算子で剰余を求めれば簡単な処理で済みます。

※2411,2412,2413,2311,2312,2313を比較対象にしても間違いではありませんが，可読性・保守性が低くなるので好ましくはありません。

※学籍番号の数値範囲を条件にするのも間違いではありませんが，可読性・保守性が低くなるほか対象外の学籍番号が含まれないように注意が必要です。

```
26      } else {
27          switch( cluster ) {
28              case 1:  printf("Ⅰ類");
29                       break;
30              case 2:  printf("Ⅱ類");
31                       break;
32              case 3:  printf("Ⅲ類");
33                       break;
34              default: printf("学籍番号が正しくありません\n");
35                       return( 3 );
36          }
37      }
38      printf(" %d年生\n", 25-year);
39
40      return( 0 );
41  }
```

※あるパラメータに対する分岐先が多い場合は可能であればswitch文を使うと可読性が向上します。

※正式な表記は「1類」「2類」「3類」ではなく「Ⅰ類」「Ⅱ類」「Ⅲ類」です。ただし，「Ⅰ」「Ⅱ」「Ⅲ」は環境によって文字化けする恐れがあるので注意が必要です。

※何の値を表示しているのかも出力したほうがユーザに混乱を与えない場合が多いです。

演習課題1-2

n個の中からr個を選ぶ組み合わせの数を求めるプログラムを作りなさい。ただし，nとrはキーボードから入力し，階乗の値は引数xに対してx!を返す単独の関数を使って計算すること。

コーディング例

```
1  /*****************************************
2     Combination
3     Oct.10,2024  prog@ds.mce.uec.ac.jp
4   *****************************************/
5
6  #include <stdio.h>
7
```

※数学関数を使うのでなければmath.hをインクルードする必要はありません。

```
 8  /*******************
 9     Factorial (n!)
10   *******************/
11
12  double fctrl(int n)
13  {
14      int     i;
15      double  x;
16
17      if( n == 0 )  return( 1.0 );
18
19      x = n ;
20      for( i=n-1 ; i>1 ; i-- ) {
21          x *= i ;
22      }
23
24      return( x );
25  }
26
```

※外部変数は保守性・可搬性を低下させることが多いので本当に必要な場合に限ったほうがよいです。

※階乗の計算はすぐに大きな値になるので，int型だと正しく扱える範囲が非常に狭くなってしまいます(文法としては間違っていません)。

※値が自然数であることを主張したい場合は，long型かlong long型を使うのがよいです。

```
27  /********
28     Main
29   ********/
30
31  int main(void)
32  {
33      int  c, n, r;
34
35      do {
36          printf("Total Number           >");
37          scanf("%d", &n);
38          printf("Number of Picking Out >");
39          scanf("%d", &r);
40      } while( n < r );
41
42      c = fctrl(n) / ( fctrl(r) * fctrl(n-r) ) ;
43      printf("%d\n", c);
44
45      return( 0 );
46  }
```

※複数の値を入力するプログラムで適切なプロンプトが出ないと，ソースコードを読むかドキュメントが用意されていなければ第三者には使えません。

※関数の戻り値を1回しか使わないのなら変数に代入する必要はありません。

※数式は「a / b / c」などとするより「a / (b * c)」など人間が書くときと同じ形で書いたほうが可読性が向上し間違いを防げます。

コーディング例2

```
 8  /********************
 9     Factorial (n!)
10   ********************/
11
12  double fctrl(int n)
13  {
14      if( n == 0 ) {
15          return( 1.0 );
16      } else {
17          return( n*fctrl(n-1) );
18      }
19  }
20
```

※このような関数を再帰関数といい，うまく使うとコードが簡潔になります。

数学関数ライブラリ

三角関数 sin, cos, tan

書　式	`#include <math.h>` `double sin(double x)`
引　数	角度 x（単位はラジアン）
戻り値	sin x の値

書　式	`#include <math.h>` `double cos(double x)`
引　数	角度 x（単位はラジアン）
戻り値	cos x の値

書　式	`#include <math.h>` `double tan(double x)`
引　数	角度 x（単位はラジアン）
戻り値	tan x の値

逆三角関数 asin, acos

書　式	#include <math.h> double asin(double x)
引　数	正弦 x (-1 ≤ x ≤ 1)
戻り値	$\sin^{-1}$ x の主値（x の単位はラジアン）

書　式	#include <math.h> double acos(double x)
引　数	余弦 x (-1 ≤ x ≤ 1)
戻り値	$\cos^{-1}$ x の主値（x の単位はラジアン）

逆三角関数 atan, atan2

書　式	`#include <math.h>` `double atan(double x)`
引　数	正接 x
戻り値	$\tan^{-1}$ x の主値（x の単位はラジアン）

書　式	`#include <math.h>` `double atan2(double y, double x)`
引　数	y：正接の分子，x：正接の分母
戻り値	$\tan^{-1}$ (y/x) の主値（x と y の単位はラジアン）

対数関数 log, log10

書　式	`#include <math.h>` `double log(double x)`
引　数	真数 x
戻り値	x の自然対数（ln x の値）

書　式	`#include <math.h>` `double log10(double x)`
引　数	真数 x
戻り値	x の常用対数（log x の値）

べき乗・平方根 pow, exp, sqrt

書　式	`#include <math.h>` `double pow(double x, double y)`
引　数	x：底，y：指数
戻り値	x^y の値

書　式	`#include <math.h>` `double exp(double x)`
引　数	指数 x
戻り値	e^x の値（e = 2.718281828…）

書　式	`#include <math.h>` `double sqrt(double x)`
引　数	実数 x
戻り値	x の正の平方根（$\sqrt{x}$ の値）

切り上げ・切り捨て ceil, floor

書　式	`#include <math.h>` `double ceil(double x)`
引　数	実数 x
戻り値	x 以上の最小の整数（型はdouble）

書　式	`#include <math.h>` `double floor(double x)`
引　数	実数 x
戻り値	x 以下の最大の整数（型はdouble）

絶対値 fabs, abs

書　式	`#include <math.h>` `double fabs(double x)`
引　数	実数 x
戻り値	x の絶対値（| x	の値）

書　式	`#include <stdlib.h>` `int abs(int x)`
引　数	整数 x
戻り値	x の絶対値（| x	の値）

数学関数を使う場合のコンパイル

math.hにある数学関数を使っているソースコードをコンパイルするときは，-lm というオプションを付けlibmをリンクすることが必要。

例 `gcc -o foo foo.c -lm`

演習課題2-1　　提出期限 10月14日(月)21時

初速 v で打ち出した球を水平方向に距離 x だけ離れた場所に落下させるには仰角 θ を何度にすればよいか求めるプログラムを作りなさい。ただし，x と v の値はプログラムの起動後に適当なプロンプトに対してキーボードから入力し，θ は 0°から90°の範囲にあるすべての値を小数点以下第1位に丸めて表示すること。

演習課題2-2　提出期限 10月14日(月)21時

実数 x に対して e^x の値を級数展開で求めるとき誤差を0.005未満にするには第何項まで展開する必要があるか求めるプログラムを作りなさい。ただし，x の値はプログラムの起動後にキーボードから入力するものとする。

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$