◆ 5 月 18 日

sample6.c(**一次元配列**)

```
#include<stdio.h>

#define N 6

int main(void)
{
  double x[N]={1.1, 2, 3.5, 4, 5.7, 6};
  double y[N]={-2, -3, -5, 6.3, 7, -1.2};
  int n=0;
  double ip=0.0;

  //x[n] の値を表示
  while(n<N){
    if(n<N-1)
      printf("x[%i]=%f\t",n,x[n]);
    else
      printf("x[%i]=%f\n",n,x[n]);  //最後は改行
    n++;
  }
  //x[n] と y[n] の内積の計算
  for(n=0; n<N; n++){
    ip += x[n]*y[n];
  }
  printf("(x,y)=%f\n",ip);

  return 0;
}
```

一次元は配列のイメージ

x[0]	x[1]	x[2]	x[3]	x[4]	x[5]

添字が 0 **から始まる** のに注意してください．もし, 勘違いして x[6] に何か値を代入しても, コンパイルの時にエラーは出ません. しかし, そのプログラムを実行するとセグメンテーション違反 (Segmentation Fault) となってプログラムが強制終了されることがあります.

sample7.c(**二次元配列**)

```
#include<stdio.h>

#define row    3
#define column 4

int main(void)
{
  int i,j;
  double a[row][column]={{1.0, 2.0, 3.0, 4.0},
                         {5.0, 6.0, 7.0, 8.0},
                         {9.0, 10.0, 11.0, 12.0}};
  double b[row][column]={{1.1, 2.2, 3.3, 4.4},
                         {5.5, 6.6, 7.7, 8.8},
                         {10.1, 11.1, 12.2, 13.3}};
  double c[row][column];

  //行列 a と行列 b の和を求める.
  printf("a+b=\n");
  for(i=0;i<row;i++){
    for(j=0;j<column;j++){
      c[i][j]=a[i][j]+b[i][j];
      printf("%5.2f\t",c[i][j]);
    }
    printf("\n");
  }

  return 0;
}
```

二次元は配列のイメージ

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

sample8.c(**関数**)

```
#include<stdio.h>

//あらかじめ使用する関数を宣言する
double f(double x);
void swap(double a,double b);

int main(void)
{
  double a=1.2,b=3.4;
  //関数 f(x) の x=a での値
  printf("f(a)=%f\n",f(a));

  //関数内での変数の入れ換え
  swap(a,b);
  printf("a=%f,b=%f in main\n",a,b);

  return 0;
}

double f(double x)
{
  return x*x+2.0*x+3.0;
}

void swap(double a,double b)
{
  double temp;

  temp=a; //a を一時保存
  a=b;    //b を a にコピーする
  b=temp; //一時保存を b にコピーする
  printf("a=%f,b=%f in swap\n",a,b);
  //void 型関数なので return はない
}
```

void 型というのは値を返さない関数です.swap 関数の中では確かに値が入れ替わっているのですが, 元の main 関数の中では入れ替わっていないことに注意してください. 関数内の変数の変化は **関数内のみで反映されます**. (ポインタというのを使えば克服できますが, 多少面倒なので省略します.)

◆ 今までのまとめとこれからの予定

5 月の下旬に坂上先生からプログラムのレポートが出ます。それまでは、Linux のコマンド、Emacs の使い方、C 言語の基本的なことを解説することになっています。

4 月 13 日、4 月 20 日までに Linux のコマンド、Emacs の使い方の最低限必要なことはできるように説明をしたつもりでいます。これからはそれらの解説はあまりせずに、C 言語の解説に集中する予定です。

今までに説明した Linux のコマンド、Emacs の使い方については http://www.math.sci.hokudai.ac.jp/~g-staff/ の「Documents」にある「Linux 基本コマンド」、「Emacs の基本キーバインド」に簡潔にまとめました。作業をするときに利用してください。さらに詳しく知りたいという方は質問をしていただくか、インターネットで検索して調べてください。