C言語講座第５回

○多次元配列

・多次元配列
int arr[3][3]のように宣言すると、配列に配列を持つことができる。（多次元配列）

```
#include <stdio.h>

int main(void)
{
        int i;
        char str[5][100];        //100文字まで入る文字列×
        for( i = 0 ; i < 5 ; i++ )
        {
                printf("文字列を入力-> ");
                scanf( "%s", str[i] );
        }
        for( i = 0 ; i < 5 ; i++ )
        {
                printf("文字列%d: %s¥n", i + 1 , str[i] );
        }
        return 0;
}
```

○画面のクリア

・system("cls")
画面に書かれている文字をクリアする。
そうでないと下にどんどん書き足されて行くことになる。
なお、このコードではちらつきが発生するが、その解決法についてはゲーム製作講座の方で。

```
#include <stdio.h>
#include <conio.h>              //kbhit()
#include <windows.h>            //system(),Sleep()

int main(void)
{
        int i,k = 0;
        char *str = "●";
        while( 1 )
        {
                for( i = 0 ; i < 30 ; i++)
                {
                        if( i == k )
                        {
                                printf("%s" , str );
                        }
                        else
                        {
                                printf("  ");
                        }
                }
                k=(k+1)%30;
                if( kbhit() )   //キーボードが押されていたら
                {
                        break;
                }
                Sleep(16);      //処理停止（ミリ秒単位）
                system("cls");  //画面のクリア
        }
        return 0;
}
```

○配列と関数（再掲）

配列を引数に持った関数
配列（文字列）を引数として関数に渡すにはその配列の最初の要素のアドレスを渡すようにする。

宣言：int function(int *arr)
使用：
int arr[5];
function(arr); //&arr[0]と同じ

この際気をつけなければいけないのは、配列をどこまで使えるか関数は知らないということ。
(a[10] と確保したのに a[15] に関数は書き込んでしまえる)
引数に配列の個数を取るなどして対処する。

#define
#define 置換前 置換後

置換前のものを置換後のものにコンパイル時に置き換える。(正しくはコンパイル前。)
右のプログラムでは
ARR_MAX を 10 に置き換える。

ファイル分割法
自作ヘッダーファイルには
関数のプロトタイプ宣言
#define、構造体（クラス）
などを書く。
関数の実際の処理などは cpp ファイルに書くことを推奨。
自作ヘッダーファイルをインクルードする場合は＜＞ではなく“”を使うこと
例：#include “mylib.h”

```
//mylib.h
//自作関数のプロトタイプ宣言
int mysort( int *arr , int arr_max);
//#define定義（左→右への置き換え。置き換えるだけ）
#define ARR_MAX 10
```

```
//mylib.cpp
#include "mylib.h"        //自作ヘッダー
//mylib.hにプロトタイプ宣言があるmysort関数の動作を書く
//配列の確保領域外への不正アクセスを防ぐため配列の要素数も受け取る
int mysort( int *arr , int arr_max)
{
        int temp;         //変数一時保存用(swap関数参照)
        int i,k;          //ループ用

        for(i = 0; i < arr_max; i++)
        {
                for( k = i+1; k < arr_max; k++)
                {
                        if( arr[i] > arr[k] )//昇順なので小さい方が先
                        {
                                temp = arr[k];
                                arr[k] = arr[i];
                                arr[i] = temp;
                        }
                }
        }
        return 0;
}
```

```
//main.cpp
#include <stdio.h> //標準のヘッダーファイル
#include "mylib.h" //自分で作ったヘッダファイルを使う

int main(void)
{
        int arr[ARR_MAX] = { 20,30,50,90,60,40,70,10,80,0};
        int i; //ループ用

        for( i = 0; i < ARR_MAX; i++)
        {
                printf("a[%d] = %d¥n", i, arr[i]);
        }

        mysort( arr , ARR_MAX);//mylib.hに宣言されている自作関数。
        printf("昇順ソートを行いました¥n");

        for( i = 0; i < ARR_MAX; i++)
        {
                printf("a[%d] = %d¥n", i, arr[i]);
        }
        return 0;
}
```

○構造体（クラス）と関数

・構造体(クラス)と関数
構造体の中の値を書き換えるために構造体をポインタで渡すときは、関数内部でどの情報を使うかを選ぶのにドットではなく、->（アロー演算子）を使う。

・構造体の初期化
宣言した順に初期化時に{ }で囲んで初期値を設定できる
宣言した順というのは
struct student {
　　char name[30];
　　int number;
}
だったら name → number の順に
struct student data = { “abc” , 1 } ;
と初期化できる。

```
#include <stdio.h>

class CPosition
{
public:
        int x , y ;
        int vx , vy ;
};
int move( CPosition* pos )
{
        pos->x += pos->vx;
        pos->y += pos->vy;
        return 0;
}
int main(void)
{
        CPosition jiki = { 0,0,4,4 };
        printf("自機の位置はx:%d y:%d です¥n",jiki.x,jiki.y);
        move( &jiki );
        printf("自機を動かしました¥n");
        printf("自機の位置はx:%d y:%d です¥n",jiki.x,jiki.y);
        return 0;
}
```

○ 演習

① ユーザー側に任意の数だけ整数値のデータを入力させてその合計を表示するプログラムを作れ。
　データの入力は０が入力されたら終了とする。

② end と入力されたら終了するプログラムを作れ。
　なお、文字列の一致判定には strcmp 関数を使ってよい（string.h をインクルードすること）
　使い方は　if(strcmp(str1 , str2) == 0)　のように使う。

③ 以下のような data.txt ファイルをつくり、そこから値を読み取って表示せよ。
　data.txt

120 45.3 hello!