

第 5 回 配列・構造体

配列

今までのように `int a,b,c` というように変数を宣言すると、変数が多い時など打ち間違いのおそれがあります。こういう同じ型の変数を宣言するときに配列を使います。

型名 配列名[配列の大きさ]

```
#include <stdio.h>

int main( void)
{
    int a[5];

    a[0]=0;
    a[1]=1;
    a[2]=2;
    a[3]=3;
    a[4]=4;
    printf( "a[0]=%d,a[1]=%d, a[2]=%d, a[3]=%d, a[4]=%d"
    ,a[0],a[1],a[2],a[3],a[4]);

    return 0;
}
```

`int a[5]={0,1,2,3,4};` もしくは `int a[]={0,1,2,3,4};`

とも代入できます。

注意すべきは配列の要素は[0]から始まっている点です。例えば `a[7]` と宣言すれば要素は `a[0]~a[6]` となります。

また for 文にすると以下のようになります。

```
#include <stdio.h>

int main( void){
    int a[]={0,1,2,3,4};
    int i;

    for(i=0;i<5;i++){
        printf( "a[%d]=%d\n", i, a[i]);
    }
    return 0;
}
```

2次元配列

2次元配列は概念的には縦×横の表であると考えられます。

```
int a[3][3]={{1,2,3},{4,5,6},{7,8,9}};
```

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

構造体

構造体は、データをひとまとまりに集めたデータ構造です。

```
struct 構造体タグ{
    型   メンバー名;
    . . . ;
};

struct タグ名 変数 1,変数 2;
```

```
#include <stdio.h>

struct xy{
    int x;
    double y;
};

int main(void){
    struct xy a,b;
    a.x=40;  a.y=50;
    b.x=60;  b.y=70;
    printf(“%d,%f\n”,a.x,a.y);
    printf(“%d,%f\n”,b.x,b.y);

    return 0;
}
```

上の時は xy が構造体タグ x,y がメンバー名 a,b が変数です。

練習問題

5人の生徒が受けた3科目(英数物)のテストの結果(整数)の科目ごとの平均値(実数)を求め表示するプログラムを作ってください。