

第5回 配列 構造体

配列

プログラムで多くの記録やデータを扱いたいときに
いままで同様 《double a, b, c・・・》や《int a1, a2, a3・・・》と
宣言すると、面倒ですしプログラムが長くなって見にくくなります。

このとき配列のテンプレ

データの型 配列名[配列の大きさ]

を使うと配列の大きさ（要素）の数ぶんまとめて宣言できます。

※ 例

int a[1], b[10], c[20];

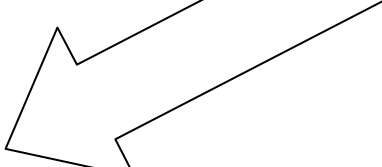
int a[4];で4個まとめて宣言

```
#include <stdio.h>
int main(void) {
    int a[4];

    a[0] = 10;
    a[1] = 20;
    a[2] = 30;
    a[3] = 40;

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

    return 0;
}
```



上のように要素1つ1つに値を代入はめんどくさいので

```
int a[4] = {10, 20, 30, 40};
```

または

```
int a[] = {10, 20, 30, 40};
```

で要素に値が代入され初期化されます。

配列で注意しなければならないのは上の式で見ればわかりますが

要素は[0]から始まるので a[5]で宣言すると要素は a[0]~a[4]になることです。

```
a[100]  
= a[0]~a[99]
```

また for 文,while 文を使うと [] に文字を使うことができます。

ただ、これは宣言するときに `int a[i];`のように宣言できる という意味ではありません。

```
#include <stdio.h>  
int main(void)  
{  
    int a[] = {10, 20, 30, 40};  
    int i;  
  
    for (i = 0; i < 4; i++)  
        printf("a[%d] = %d ¥n", i, a[i]);  
  
    return 0;  
}
```

おまけ 2次元配列

配列の延長のようなものです。

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

これで初期化

$a[0][0]=1, a[0][1]=2 \sim a[2][0]=7 \sim$

※例

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]

行列の座標と逆!!

構造体

構造体も配列同様にデータをまとめるときに用います

構造体のテンプレ

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

struct タグ名 変数 1,変数 2;
```

型 = int,double
メンバー名 = 変数、配列

メンバーを指定するときは、

変数.メンバー名

とドットを使い指定します。

構造体では別の型(int 型 double 型 etc)もまとめることができます。

```

#include <stdio.h>
struct spec{
    int tall;
    double weight;
};

int main(void){
    struct spec a,b;
    a.tall = 156;  a.weight = 50;
    b.tall = 170;  b.weight = 60;
    printf("A 君(%dcm,%fkg)¥n", a.tall,  a.weight);
    printf("B 君(%dcm,%fkg)¥n", b.tall,  b.weight);

    return 0;
}

```

変数.メンバー名で指定

この場合 spec が構造体タグ tall,weight がメンバー名 a,b が変数になります。

メンバー名 変数名に配列を使うとき

int x[5]; とか struct spec a[10];

と宣言することになります。

この場合の注意点として変数やメンバー名を指定するときに

[]を忘れないようにしなければいけません。

上の場合

a[0].tall b[3].weight a.tall[4] b[3].weight[5]

のようになります。

演習問題

3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