

C言語講座

第四回

配列

▶ 配列とは？

- ▶ 同種の型のデータ(intやdouble等)をまとめて格納できるもの
- ▶ 一つの変数が箱なら配列はタンスのようなイメージ
- ▶ forループ文で順番に参照させるのが楽



配列 2

▶ 定義

▶ 型 配列変数名[要素数];

- ▶ 例: `int i[10];`
- ▶ 上記の宣言の場合の配列は `i[0]~i[9]` となっている
`i[1]~i[10]` ではないことに注意。

▶ 初期化

- ▶ 例: `int i[3] = {11,2,4};`
- ▶ `int i[3] = {3};`
上記のように宣言すると `{3,0,0}` と格納される。
- ▶ また、初期化データを宣言する場合は `[]` の中 (要素数) を省略することもできる
- ▶ `int i[] = {4,1};` と `i[2] = {4,1};` は結果的に同じ
- ▶ `[]` で囲んで記述する数字を「添え字」という。



例文

【ソース】

```
#include<stdio.h>

int main(void)
{
    int i[10] = {2};
    int num;

    // i[num]の二乗を次の添字のデータに入れる
    for(num=0; num<10; num++){
        printf("%5d", i[num]);
        i[num+1] = i[num]+i[num];
    }
    return 0;
}
```

【出力】

2 4 8 16 32 64 128 256 512 1024



練習

- ▶ 順番の入れ替え
- ▶ 添字番号0~5の配列の中に入ってる数字の順番を逆にさせてみよう

$i[0]=0 \rightarrow i[0]=5$

$i[1]=1 \rightarrow i[1]=4$

...

$i[5]=5 \rightarrow i[5]=0$



多次元配列

▶ 定義

▶ 型 配列変数名[縦の大きさ][横の大きさ];

▶ 例: `int i[2][3] = { {1, 2}, {3, 4}, {5, 6} };`

▶

1	2
3	4
5	6

▶ 縦＝行、横＝列として行列をイメージすると良い



例文

【ソース】

```
{  
    int a[2][2] = {{1,2},{3,4}};  
    int b[2][2] = {{2,3},{4,5}};  
    int ans[2][2];  
    int x,y; //行、列  
    for(x=0; x<2; x++){  
        for(y=0; y<2; y++){  
            ans[x][y] = (a[x][y] + b[x][y]);  
        }  
    }  
    printf(“%3d%3d¥n%3d%3d¥n”,ans[0][0],ans[0][1],ans[1][0],ans[1][1]);  
    return 0;  
}
```

【結果】

3 5

7 9



型について

- ▶ 整数のみを扱うint型や文字列を扱うchar型など
- ▶ 代表的なもの

型	データ	バイト幅	数値範囲
char	文字型	1	-128～-127
int	整数型	2	-32768～32767
long	倍長整数型	4	-2147483648～ 2147483647
float	単精度浮動小数点型	4	省略
double	倍精度浮動小数点型	8	省略

- ▶ 型により扱えるデータ型や数値制限があるので興味があったら調べてね

練習（余裕のある人向け）

- ▶ 多次元配列を使って 2×2 の行列の掛算
- ▶ キーボードから入力された8つの整数をそれぞれ配列に格納していき、その出力をだすプログラムを作ってみよう
- ▶ ヒント: 3重ループ



答え

```
int main(void)
{
    //変数宣言
    int a[2][3]={0};
    int b[3][2]={0};
    int ans[2][2] = {0};
    int i,j,k,temp; //tempは和を一時保存
    //格納
    printf("行列a\n");
    for(i=0;i<2;i++){
        for(j=0;j<3;j++){
            printf("%d行%d列目 :",i,j);
            scanf("%d",&a[i][j]);
        }
    }
    printf("行列b\n");
    for(i=0;i<3;i++){
        for(j=0;j<2;j++){
            printf("%d行%d列目 :",i,j);
            scanf("%d",&b[i][j]);
        }
    }
}
```

```
    //計算
    for(i=0;i<2;i++){
        for(j=0;j<2;j++){
            temp = 0;
            for(k=0;k<3;k++){
                temp += a[i][k]*b[k][j];
            }
            ans[i][j] = temp;
        }
    }
    //表示
    printf("%3d%3d\n%3d%3d\n",ans[0][0],ans[0][1],ans[1][0],ans[1][1]);
    return 0;
}
```