

## 基礎課題1 解答例:

文字配列を画面に表示するプログラム.

```
main(void)
```

```
{ char str[] = "This is a string!";
```

```
    printf("%c¥n", str[①]); → 解答: 0
```

```
    printf("%c¥n", str[②]); → 解答: 16
```

```
    printf("%c¥n", str[③]); → 解答: 4, 7, 9, 17
```

```
    return 0;
```

```
}
```

画面表示結果

T  
!

← 何も表示されない

答えは4つあります

## 基礎課題2解答例:

double型配列A(要素数3)に,  
1.23, 2.34, 3.45の値を初期設定し, それらの  
合計値を画面表示するプログラムを作成せよ

```
main(void)
{
    double A[3] = {1.23, 2.34, 3.45};
    printf("%f¥n", A[0]+A[1]+A[2]);
    return 0;
}
```

# 特講I課題1 変数の宣言、値の範囲

- 以下の6つの数値を保存するために必要な変数を宣言して、各数値を初期値として設定しなさい。また、printf関数を用いて各変数の値を画面に表示しなさい。ただし、変数は指示された値を保存するために最小限のメモリ領域を使用するように型を選択しなさい。(例えば、12という値を保存するための変数はint型ではなくchar型を利用する)
- 127、200、300、123456、1234.5、12345678901234

## プログラム例

```
#include <stdio.h>
```

```
main()
```

```
{
```

```
    char          c1  = 127;
```

```
    unsigned char c2  = 200;
```

```
    short int     si1 = 300;
```

```
    int           i1  = 123456;
```

```
    float         f1  = 1234.5;
```

```
    long int      li1 = 12345678901234;
```

```
    double        d1  = 12345678901234.0;
```

```
    printf("c1 = %d ¥n", c1);
```

```
    printf("c2 = %u ¥n", c2);
```

```
    printf("si1 = %d ¥n", si1);
```

```
    printf("i1 = %d ¥n", i1);
```

```
    printf("f1 = %f ¥n", f1);
```

```
    printf("li1 = %ld ¥n", li1);
```

```
    printf("d1 = %f ¥n", d1);
```

```
}
```

long int型は8byteで扱える値の範囲は、  
-9,223,372,036,854,775,808 ~  
9,223,372,036,854,775,807

} いずれも8byte

桁が大きな整数値を扱うにはlong int型かdouble型を利用する必要があり、printfで画面表示するにはlong intでは%ld、double型では%fを利用する必要がある。

## 特講I課題2 配列の宣言

- 配列名がaのchar型の配列を宣言し、初期値として1,2,4,8,16,32,64,128を設定しなさい。配列の要素を要素番号の小さい順に表示しなさい。また、すべての要素の総和も求めなさい。

<注意>

- char型の値の範囲を超えるものは、正しく画面に表示されない。

### プログラム例

```
#include <stdio.h>
main()
```

```
{
```

```
    char a[]={1,2,4,8,16,32,64,128};
```

```
    printf("%d¥n", a[0]);
```

```
    printf("%d¥n", a[1]);
```

```
    printf("%d¥n", a[2]);
```

```
    printf("%d¥n", a[3]);
```

```
    printf("%d¥n", a[4]);
```

```
    printf("%d¥n", a[5]);
```

```
    printf("%d¥n", a[6]);
```

```
    printf("%d¥n", a[7]);
```

```
    printf("%d¥n", a[0]+a[1]+a[2]+a[3]+a[4]+a[5]+a[6]+a[7]);
```

```
    return 0;
```

```
}
```

### 実行結果例

1

2

4

8

16

32

64

-128 ← オーバーフロー

-1

## 特講I課題3 配列の宣言2

- 課題2の配列の型をunsigned char型に変えて実行した結果を示しなさい。

### プログラム例

```
#include <stdio.h>
main()
{
    unsigned char a[]={1,2,4,8,16,32,64,128};
    printf("%u¥n", a[0]);
    printf("%u¥n", a[1]);
    printf("%u¥n", a[2]);
    printf("%u¥n", a[3]);
    printf("%u¥n", a[4]);
    printf("%u¥n", a[5]);
    printf("%u¥n", a[6]);
    printf("%u¥n", a[7]);
    printf("%u¥n", a[0]+a[1]+a[2]+a[3]+a[4]+a[5]+a[6]+a[7]);
    return 0;
}
```

### 実行結果例

```
1
2
4
8
16
32
64
128
255
```

## 特講I課題4 8進数の表現

- 課題3の配列の初期値{1,2,4,8,16,32,64,128}を8進数の表現に変えたプログラムを作成するとともに、実行結果が課題3と一致することを確認しなさい。

### プログラム例

```
#include <stdio.h>
main()
{
    unsigned char a[]=
        {01,02,04,010,020,040,0100,0200};
    printf("%u¥n", a[0]);
    printf("%u¥n", a[1]);
    printf("%u¥n", a[2]);
    printf("%u¥n", a[3]);
    printf("%u¥n", a[4]);
    printf("%u¥n", a[5]);
    printf("%u¥n", a[6]);
    printf("%u¥n", a[7]);
    printf("%u¥n", a[0]+a[1]+a[2]+a[3]+
        a[4]+a[5]+a[6]+a[7]);
    return 0;
}
```

10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

## 特講I課題5 16進数の表現

- 課題3の配列の初期値{1,2,4,8,16,32,64,128}を16進数の表現に変えたプログラムを作成するとともに、実行結果が課題3と一致することを確認しなさい。

### プログラム例

```
#include <stdio.h>
main()
{
    unsigned char a[]=
        {0x1,0x2,0x4,0x8,0x10,0x20,0x40,0x80};
    printf("%u¥n", a[0]);
    printf("%u¥n", a[1]);
    printf("%u¥n", a[2]);
    printf("%u¥n", a[3]);
    printf("%u¥n", a[4]);
    printf("%u¥n", a[5]);
    printf("%u¥n", a[6]);
    printf("%u¥n", a[7]);
    printf("%u¥n", a[0]+a[1]+a[2]+a[3]+
        a[4]+a[5]+a[6]+a[7]);
    return 0;
}
```

10進数	2進数	8進数	16進数
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10

# 特講I課題6 文字配列の宣言と初期化1

データ型 配列名[ ] = {定数, ..., 定数};

- 上記形式にて、文字配列(配列名: name)を宣言し、自分の名前の後ろに?(クエスチョンマーク)をつけた文字列を初期値として与えるようにプログラムを記述せよ。初期値として与えた文字列を画面に表示するプログラムと実行結果を示しなさい。

## プログラム解答例

```
#include <stdio.h>
main()
{
    char name[ ] = {'s', 'u', 'g', 'a', '¥?', '¥0'};
    printf("name = %s ¥n", name);
}
```

## 実行例

name = suga?



# 特講I課題7 文字コードの表示

- 課題6にて作成した文字配列の全ての文字(先頭の文字からヌル文字まで)の文字と文字コードを1文字1行ずつ画面に表示するプログラムを作成し、プログラムと実行結果を示しなさい。
- 文字と文字コードの関係が正しいことを教科書で確認すること。

## プログラム解答例

```
#include <stdio.h>
main()
{
    char name[] = {'s', 'u', 'g', 'a', '¥?', '¥0'};
    printf("%c = %d ¥n", name[0], name[0]);
    printf("%c = %d ¥n", name[1], name[1]);
    printf("%c = %d ¥n", name[2], name[2]);
    printf("%c = %d ¥n", name[3], name[3]);
    printf("%c = %d ¥n", name[4], name[4]);
    printf("%c = %d ¥n", name[5], name[5]);
}
```

## 実行例

```
s = 115
u = 117
g = 103
a = 97
? = 63
= 0
```

# 特講I課題8 文字配列の宣言と初期化2

`char name2[] = "自分の名前";`

- 上記形式にて、自分の名前を初期値として与えるようにせよ。ただし、「名字」と「名前」の間にはタブ文字をエスケープシーケンスで加えなさい。初期値として与えた文字列を画面に表示するプログラムと実行結果を示しなさい。

## プログラム解答例

```
#include <stdio.h>
main()
{
    char name2[] = "Suga\tMikio";
    printf("name = %s ¥n", name2);
}
```

タブ文字



## 実行例

name = Suga Mikio

## 特講I課題9 文字配列の操作

- 課題8にて作成したプログラムに、「タブ文字を改行文字に上書きする代入文」と、「文字列の内容を表示するprintf()文」の2行だけ追加することで、下記例のように画面表示の2行目に「名字」、3行目に「名前」が表示されるようにしたプログラムを示しなさい。

### プログラム解答例

```
#include <stdio.h>
main()
{
```

```
    char name2[] = "Suga¥tMikio";
    printf("name = %s ¥n", name2);
```

```
    name2[4] = '¥n'; ←名字の文字数によって要素番号を変える必要あり
    printf("%s ¥n", name2);
```

```
}
```

### 実行例

```
name = Suga   Mikio
Suga
Mikio
```

# 特講I課題10 文字配列の操作

- 課題8にて作成したプログラムに、`#include <string.h>` と「strcpyを利用した文」、「文字列の内容を表示するprintf()文」の3行だけを追加することで(プログラムの他の箇所には変更を加えないで)、画面表示の2行目に「名字」、3行目に「名前」が表示されるようにしたプログラムと実行結果を示しなさい。

## プログラム解答例

```
#include <stdio.h>
#include <string.h>
main()
{
    char name2[] = "Suga¥tMikio";
    printf("name = %s ¥n", name2);
    strcpy(name2, "Suga¥nMikio");
    printf("¥s ¥n", name2);
}
```

## 実行例

```
name = Suga   Mikio
Suga
Mikio
```