

一次元配列

A 一次元配列

一次元配列の宣言は、次の書式で記述する。

データ型 配列名 [要素の個数] ;

【例】 `int a[5];`

この例では、a という名前の整数型配列が以下のように確保される。

a[0]	a[1]	a[2]	a[3]	a[4]
------	------	------	------	------

宣言時に指定したのは要素の個数であり、プログラム中で配列を使用する場合には、0～4（要素の個数－1）番の要素を扱うことにじゅうぶん注意すること。

B 二次元配列

二次元配列の宣言は、次の書式で記述する。

データ型 配列名 [要素の個数] [要素の個数] ;

【例】 `int b[3][4];`

この例では、b という名前の整数型配列が以下のように確保される。

b [0] [0]	b [0] [1]	b [0] [2]	b [0] [3]
b [1] [0]	b [1] [1]	b [1] [2]	b [1] [3]
b [2] [0]	b [2] [1]	b [2] [2]	b [2] [3]

一次元配列 box[]

```
#include <stdio.h>
int main(void)
{
    int box[5]={786,386,831,23,865};
    for (int i=0;i<5;i++){

        printf("box[%d]=%d¥n",i,box[i]);

    }
    return 0;
}
```

結果

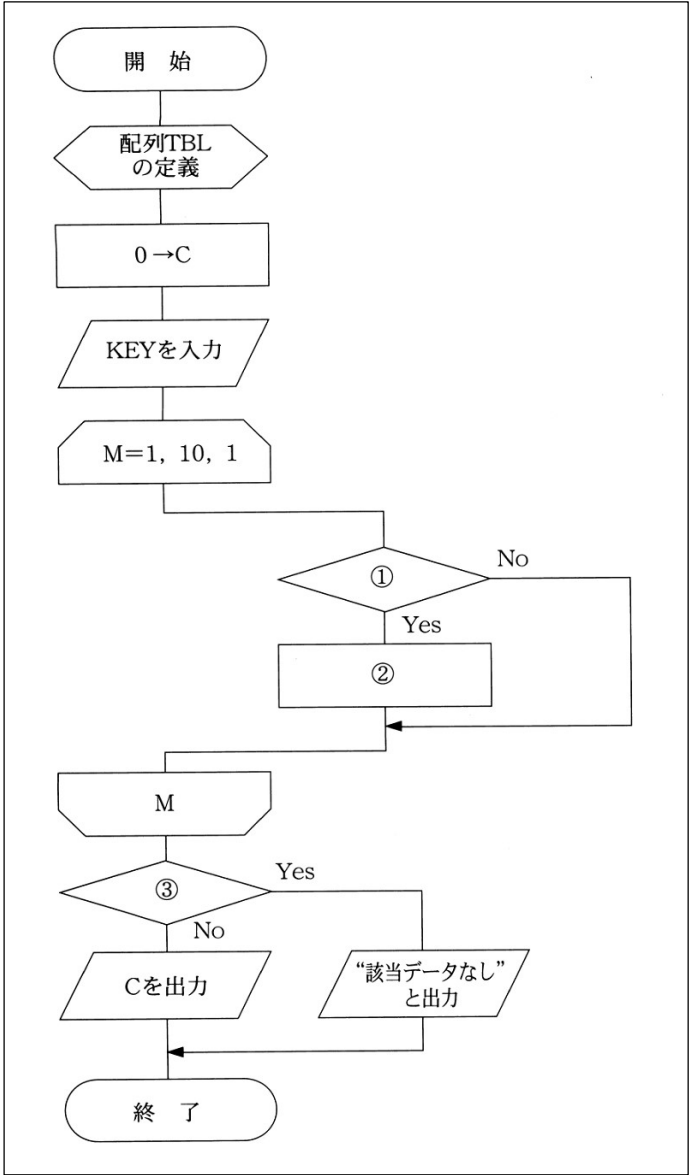
```
box[0]=786
box[1]=386
box[2]=831
box[3]=23
box[4]=865
```

```
#include <stdio.h>
int main(void)
{
    int i,key;
    int box[5]={786,386,831,23,865};
    printf("key 数值入力");
    scanf("%d",&key);
    for (i=0;i<5;i++){
        if(box[i]==key){
            printf("i=%dbox[%d]%d= %d¥n",i,i,box[i],key);
        }else{
            printf("i=%dbox[%d]%d<> %d¥n",i,i,box[i],key);
        }
    }
    return 0;
}
```

key 数值入力 23
i=0 box[0]786 <> 23
i=1 box[1]386 <> 23
i=2 box[2]831 <> 23
i=3 box[3]23 = 23
i=4 box[4]865 <> 23

key 数值入力 333
i=0 box[0]786 <> 333
i=1 box[1]386 <> 333
i=2 box[2]831 <> 333
i=3 box[3]23 <> 333
i=4 box[4]865 <> 333

次の流れ図は、データ KEY を入力し、KEY と同じ値が配列 TBL にいくつあるかを出力するものである。また、配列 TBL になければ、“該当データなし”と出力する。流れ図中の①～③に最も適する語句を解答群から選び、記号で答えなさい。



配列の内容

TBL(1)	3 1
TBL(2)	5 2
TBL(3)	6 2
TBL(4)	7 5
TBL(5)	3 2
TBL(6)	8 8
TBL(7)	2 6
TBL(8)	9 3
TBL(9)	7 5
TBL(10)	4 9

解答群

- ア M=KEY
- イ TBL(M) ≠ KEY
- ウ TBL(M)=KEY
- エ C=0
- オ C≠
- カ C+M→C
- キ C+1→C
- ク C+TBL(M)→C