

Chương 6. MẢNG

1. Giới thiệu
2. Mảng – Các ví dụ về mảng
3. Truyền tham số cho mảng
4. Sắp xếp mảng
5. Tìm Mean, median, mode sử dụng mảng
6. Tìm dữ liệu của mảng
7. Mảng 2 chiều

1. Giới thiệu

- Mạng là tập hợp các phần tử có cùng dữ liệu
- Có cấu trúc tĩnh – Cùng kích thước xuyên suốt chương trình
- Một nhóm các địa chỉ nhớ liên tiếp

2. Mảng

- Để tham chiếu đến một phần tử, xác định

Tên mảng

Vị trí

- Định dạng:

Ten_mang [vi_tri]

- Tên của các phần tử của mảng c có n phần tử là

$c[0], c[1], \dots, c[n-1]$

- Định nghĩa mảng

Kieu_du_lieu ten_mang[so_phan_tu]

VD: `int c[10]`

- Khởi tạo mảng

`int n[5] = {1,2,3,4,5}`

`int n[5] = {0}`

`int n[] = {1,2,3,4,5}`

Nhập và đọc dữ liệu mảng

- Nhập dữ liệu:

```
for (i = 0; i < 10; i++)  
{  
    printf("enter element %d\n", i);  
    scanf("%d", &s[i]);  
}
```

- Đọc dữ liệu từ mảng:

```
for (i = 0; i < 10; i++)  
{printf("%7d%13d\n", i, n[i]);
```

VD mảng

1. Nhập và xuất 1 mảng 10 phần tử theo cột gồm cột tên phần tử và giá trị
2. Nhập và xuất 1 mảng 10 phần tử theo cột với các phần tử là 0
3. Khởi tạo 1 mảng 10 phần tử và in ra theo cột
4. Sử dụng định nghĩa size để khai báo mảng, giá trị mảng $s[i] = 2+2i$ xuất theo cột
5. Nhập, xuất các phần tử của mảng 10 phần tử và tính tổng các phần tử
6. In số hình ngôi sao tương ứng với dữ liệu cho trước trong mảng
7. Thống kê kết quả khảo sát

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

```
#include <conio.h>
```

```
int main(void){
```

```
    int a[10];
```

```
    int i;
```

```
    for (i = 0; i < 10; i++){
```

```
        printf("enter value %d\n", i + 1);
```

```
        scanf("%d", &a[i]);
```

```
    }
```

```
    printf("%s%10s\n", "element", "value");
```

```
    for (i = 0; i < 10; i++)
```

```
        printf("%4d%10d\n", i+1, a[i]);
```

```
    getch();
```

```
}
```

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

```
#include <conio.h>
```

```
int main(void)
```

```
{
```

```
    int n[10];
```

```
    int i;
```

```
    for (i = 0; i < 10; i++)
```

```
        {n[i] = 0;}
```

```
    printf("%s%13s\n", "Element", "Value");
```

```
    for (i = 0; i < 10; i++)
```

```
        {printf("%7d%13d\n", i, n[i]);}
```

```
    getch();
```

```
}
```

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    int n[10] = { 32, 27, 64, 18, 95, 14, 90,
70, 60, 37 };
    int i;
    printf("%s%13s\n", "Element", "Value");
    for (i = 0; i < 10; i++)
    {
        printf("%7d%13d\n", i, n[i]);
    }
    getch();
}
```



```
#include <stdio.h>
#include <conio.h>
#define size 10
int main(void)
{
    int s[size];
    int j;
    for (j = 0; j < size; j++)
        {s[j] = 2 + 2 * j;}
    printf("%s%13s\n", "Element", "Value");
    for (j = 0; j < size; j++)
        {printf("%7d%13d\n", j, s[j]);}
    getch();
}
```

```
#include <stdio.h>
#include <conio.h>
#define size 10
int main(void){
    int s[size];
    int sum = 0,j;
    for (j = 1; j <= size; j++){
        printf("enter element %d\n", j);
        scanf("%d", &s[j]);
    }
    printf("s[%d] = {",size);
    for (j = 1; j <= size; j++){
        sum = sum + s[j];
        if (j == (size))
            printf("%3d", s[j]);
        else
            printf("%3d,", s[j]);
    }
    printf("}\n Sum of all elements is = %d ",sum);
    getch();
}
```

```
#include <stdio.h>
#include <conio.h>
#define SIZE 10
int main(void)
{
    int n[SIZE] = { 19, 3, 15, 7, 11, 9, 13, 5, 17, 1 };
    int i, j;
    printf("%s%13s%17s\n", "Element", "Value", "Histogram");
    for (i = 0; i < SIZE; i++)
    {
        printf("%7d%13d", i, n[i]);
        for (j = 1; j <= n[i]; j++)
            { printf("%c", '*'); } // printf("*");
        printf("\n");
    }
    getch();
}
```

```
#include <stdio.h>
#include <conio.h>
#define RES_S 40
#define FRE_S 11
int main(void){
    int answer,rating;
    int fre[FRE_S] = { 0 };
    int responses[RES_S] = { 1, 2, 6, 4, 8, 5, 9, 7,
8, 10,1, 6, 3, 8, 6, 10, 3, 8, 2, 7, 6, 5, 7, 6, 8, 6,
7, 5, 6, 6,5, 6, 7, 5, 6, 4, 8, 6, 8, 10 };
    for (answer = 0; answer < RES_S; answer++)
        ++fre[responses[answer]];
    printf("%s%17s\n", "Rating", "Frequency");
    for (rating = 1; rating < FRE_S; rating++)
        printf("%6d%17d\n", rating, fre[rating]);
    getch();
}
```

- Tính số lần xuất hiện:

+ Cách 1: sử dụng switch, if

+ Sử dụng mảng

VD: A [14] = { 4 , 1 , 6 , 2 , 1 , 5 , 6 , 4 , 2 , 3 , 1 , 3 , 5 , 6 }

B[7] = {0 } = { 0,0,0,0,0,0,0}

```
for ( int i = 0; i<=13; i++)  
    ++B[A[i]];
```

```
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 7
int main( void ) {
    size_t face; // random die value 1 - 6
    unsigned int roll; // roll counter 1-6,000,000
    unsigned int frequency[ SIZE ] = { 0 }; // clear
counts
    srand( time( NULL ) );
    for ( roll = 1; roll <= 6000000; ++roll ) {
        face = 1 + rand() % 6; // end for
        ++frequency[face];
    }
    printf( "%s%17s\n", "Face", "Frequency" );
    for ( face = 1; face < SIZE; ++face ) {
        printf( "%4d%17d\n", face, frequency[ face ] );
    }
    _getch();
}
```

Mảng ký tự

- Khai báo mảng ký tự: `char string1[] = "first";`

Trong đó có 5 ký tự (f-i-r-s-t) và 1 ký tự đặc biệt “\0” – Ký tự kết thúc

→ có 6 ký tự

- Có thể khởi tạo cách khác:

```
char string1[ ] = { 'f', 'i', 'r', 's', 't', '\0' };
```

- Có thể tham chiếu đến một ký tự của mảng

`string1[3]` là ký tự ‘s’

- Khi sử dụng đọc mảng ký tự

```
scanf( "%s", string2 );
```

Đọc dữ liệu đến khi gặp phải tab, khoảng trắng, enter, eof (end of file)

VD. In chuỗi được nhập, chuỗi đã được khởi tạo, cho khoảng trắng vào chuỗi đã nhập

```
#include <stdio.h>
#include <conio.h>
int main(void){
    char string1[20];
    char string2[] = "string example";
    int i;
    printf("Enter a string: ");
    gets_s(string1);
    printf("string1 is: %s\nstring2 is: %s\n"
"string1 with spaces between characters is:\n",
string1, string2);
    for (i = 0; string1[i] != '\0'; i++){
        printf("%c ", string1[i]);
    }
    printf("\n");
    _getch();
}
```


Cho nhập chuỗi, Nhập ký tự cần tìm c. In ra có bao nhiêu ký tự c trong chuỗi

```
int main(void){
    char string1[20];
    int j = 0 , i; char c;
    printf("Enter a string: ");
    gets_s(string1);
    printf("Enter a character: ");
    scanf_s("%c",&c);
    for (i = 0; string1[i] != '\0' ; i++){
        if (string1[i] == c)    j++;
    }
    if (j != 0)
        printf("Have %d character %c in string",j,c);
    else printf("Have no character %c in string", c);
}
```

```
for (i = 0; i <= strlen(string1); i++){  
    if (string1[i] == c)  
        j++;  
}
```

`strlen()`; nằm trong thư viện `#include <string.h>`

3. Truyền mảng cho hàm

- Truyền mảng
- Truyền tên mảng, bỏ dấu []

VD: truyền mảng array1 đến hàm function1

```
int array1[50];  
function1(array1, 50);
```

Kích thước mảng thường được truyền nhưng không nhất thiết

- Mảng truyền tham chiếu
- Tên của mảng là địa chỉ của phần tử đầu tiên
- Hàm sẽ biết nơi nào mảng được lưu trữ
- Truyền phần tử mảng – truyền tham trị
- Truyền tên phần tử (VD: array1[1]) đến hàm

3. Truyền mảng cho hàm

- Hàm nguyên mẫu:

```
void modifyArray( int b[], int arraySize );
```

- Trong đó:

`int b[]` có thể được viết `int []`

`int arraySize` có thể được viết `int`

VD: Vị trí mảng là vị trí đầu tiên của phần tử

```
#include <stdio.h>
#include <conio.h>
int main(void)
{
    char array[5];
    printf("    array = %p\n&array[0] = %p\n
    &array = %p\n", array, &array[0], &array);
    getch();
}
```