



Chương 6

KIỂU MẢNG (ARRAY TYPE)

Nội dung



Giới thiệu



Mảng 1 chiều



Mảng 2 chiều



Bài tập



Chương 6 KIỂU MẢNG (ARRAY TYPE)

Giới thiệu

❑ Nội dung trong chương này:

- Giới thiệu về mảng dữ liệu
- Giới thiệu về cách truy xuất/thay đổi phần tử trong mảng dữ liệu
- Giới thiệu về cách kiểm tra một phần tử có thuộc mảng dữ liệu hay không

❑ Bạn sẽ có thể:

- Thực hiện các phép thao tác trên chuỗi như tạo mảng, thay đổi giá trị, truy xuất phần tử, kiểm tra phần tử thuộc mảng trong ngôn ngữ C/C++

Chương 6 KIỂU MẢNG (ARRAY TYPE)

Giới thiệu

- Ví dụ 1 : bạn muốn tính trung bình 10 số nguyên nhập vào từ bàn phím, bạn sẽ khai báo 10 biến: a, b, c, d, e, f, g, h, i, j có kiểu int và lập thao tác nhập cho 10 biến.
- Điều này chỉ phù hợp với n nhỏ, còn đối với n lớn thì khó có thể thực hiện được. Vì vậy khái niệm mảng được sử dụng

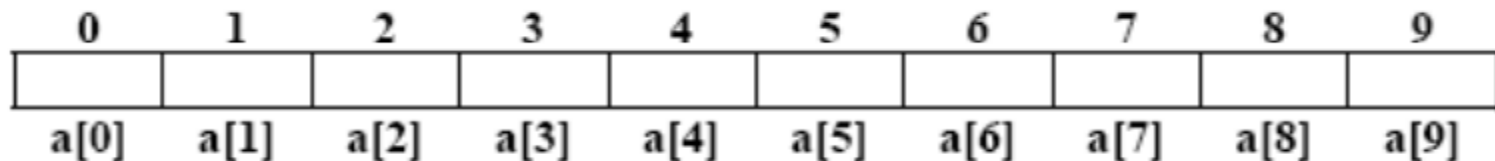
II-MẢNG MỘT CHIỀU

1- Khái niệm:

Mảng 1 chiều là tập các biến nhớ cùng kiểu, mỗi phần tử được xác định bởi tên mảng và một chỉ số bắt đầu từ 0 trở đi.

1. Khái niệm

- Mảng là một tập hợp các biến có cùng kiểu dữ liệu nằm liên tiếp nhau trong bộ nhớ và được tham chiếu bởi một tên chung chính là tên mảng.
- Mỗi phần tử của mảng được tham chiếu thông qua chỉ mục (index)



1. Khái niệm

- Nếu mảng có n phần tử thì phần tử đầu tiên có chỉ mục là 0 và phần tử cuối có chỉ mục là $n-1$.
- Để tham chiếu đến một phần tử ta dùng tên mảng và chỉ mục của phần tử được đặt trong cặp dấu [].
 - Ví dụ: `a[0]`
- Số phần tử trong mảng được gọi là kích thước của mảng. luôn cố định, phải được xác định trước và không đổi trong suốt quá trình thực hiện chương trình.

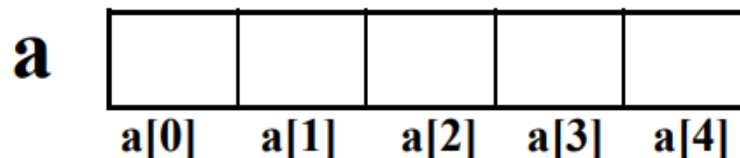
2. Mảng một chiều

- Khai báo một mảng một chiều
- Cú pháp:

`type arrayName[elements];`

- type: kiểu dữ liệu của mỗi phần tử mảng.
- elements: số phần tử có trong mảng
- arrayName: tên mảng

Ví dụ: `int a[5]`



2. Mảng một chiều

- Mảng phải được khai báo tường minh
- Kích thước (tính bằng byte) của mảng được tính theo công thức:

$$\text{Total_size} = \text{sizeof}(\text{type}) * \text{elements}$$

Ví dụ: `int num[100];`

Mảng `num` có kích thước là:

`2bytes * 100 = 200bytes` (giả sử `int` chiếm 2 bytes)

2.Mảng một chiều

- Mỗi phần tử mảng là một biến thông thường.

Ví dụ: `int num[3];`

`num[0] = 2; //gán 2 cho phần tử num[0]`

`num[1] = num[0] + 3 //num[1] có giá trị 5`

`num[2] = num[0] + num[1]; //num[2] có giá trị 7`

`cout << num[1]; // In ra giá trị 5`

2. Mảng một chiều

- Khai báo và khởi tạo mảng một chiều

- Cú pháp:

`type arrayName[] = {value1, value2, ..., valuen};`

- Lưu ý:

- ✓ Không khai báo kích thước mảng.
- ✓ Số lượng phần tử trong mảng là số các giá trị được cung cấp trong cặp dấu ngoặc {}, được phân cách nhau bởi dấu phẩy.

2. Mảng một chiều

Ví dụ: Tạo một mảng nguyên a có N phần tử. Mỗi phần tử có giá trị là chỉ mục của nó. In mảng ra màn hình.

```
#include <iostream.h>
#include <conio.h>
#define N 10
void main()
{
    int a[N];
    for(int i=0 ; i < N ; i++)
        a[i] = i ;
    cout<< "In mang:\n";
    for(int i=0 ; i < N ; i++)
        cout << "a[" << i <<"] = " << a[i] <<
        endl;
}
```

2. Mảng một chiều

Ví dụ: Đổi một số nguyên dương thập phân thành số nhị phân

```
void main()
```

```
{
```

```
    int i,j=0, n, np[20];
```

```
    cout<<"n="; cin>>n;
```

```
    do
```

```
    {
```

```
        np[j]= n%2;
```

```
        j++;
```

```
        n = n/2;
```

```
    }while(n>0);
```

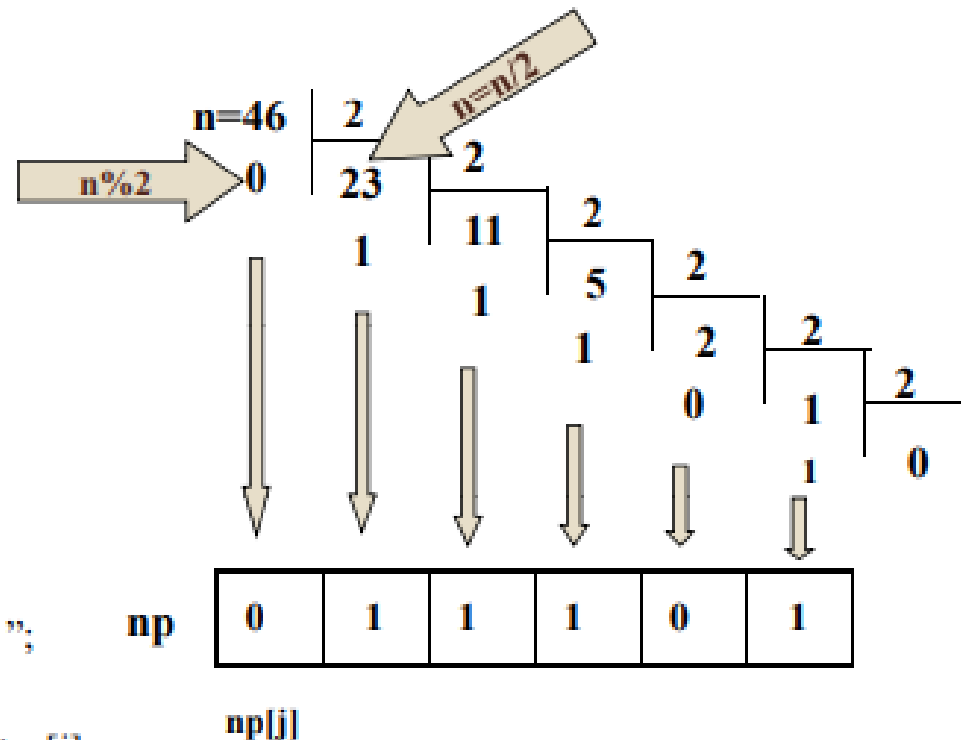
```
    cout<<"dạng nhị phân: ";
```

```
    for(i=j-1 ; i>0 ; i--)
```

```
        cout<<setw(3)<<np[i];
```

```
    getch();
```

```
}
```



2. Mảng một chiều

Ví dụ : Nhập vào một mảng số nguyên sau đó sắp xếp theo thứ tự tăng dần

```
#include <iostream.h>
#define n 5
main ( )
{
    int a [ n ] ; int i , j, t ;
    for ( i = 0 ; i < n ; i ++ ) // nhập mảng
    { cout << "a [" << i << " ] = "; cin >> a[i]; cout << endl; }
    for ( i = 0 ; i < n - 1 ; i ++ ) // sắp xếp
        for ( j = i + 1 ; j < n ; j ++ )
            if ( a [ i ] > a [ j ] )
                { t = a [ i ] ; a [ i ] = a [ j ] ; a [ j ] = t ; }
    for ( i = 0 ; i < n ; i ++ ) // xuất mảng
        cout << setw(3) << a[i];
    getch ( ) ;
}
```

2.Mảng một chiều

- Sử dụng hàm tạo số ngẫu nhiên
 - ✓ C++ cung cấp hàm random để tạo ra các số ngẫu nhiên.
Cú pháp: `int random(int n)`
 - ✓ Kết quả của hàm là tạo ra các số nguyên ngẫu nhiên từ 0 đến $n-1$
 - ✓ Khi sử dụng random ta phải gọi randomize để khởi tạo chế độ tạo số ngẫu nhiên.
 - ✓ Để sử dụng các hàm trên thì trong chương trình phải khai báo thư `<stdlib.h>`

2. Mảng một chiều

Ví dụ: tạo mảng ngẫu nhiên và in ra màn hình.

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
#include <stdlib.h>
void main()
{
    randomize();
    cout << "Tạo mảng ngẫu nhiên :\n";
    for (int i=0; i<n; i++)
        a[i]=random(100);
    for(int j=0; j<n; j++)
        cout << setw(3) << a[j];
}
```

2. Mảng một chiều

Ví dụ: tạo mảng ngẫu nhiên và in ra màn hình.

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>
#include <stdlib.h>
void main()
{
    randomize();
    cout << "Tạo mảng ngẫu nhiên :\n";
    for (int i=0; i<n; i++)
        a[i]=random(100);
    for(int j=0; j<n; j++)
        cout << setw(3) << a[j];
}
```


3. Mảng nhiều chiều

- C/C++ hỗ trợ mảng nhiều chiều.
- Dạng đơn giản nhất của mảng nhiều chiều là mảng hai chiều.
- Mảng hai chiều thực chất là mảng của những mảng một chiều. Ta có thể xem mảng hai chiều là một ma trận gồm các hàng và các cột

3. Mạng nhiều chiều

- Khái niệm mạng hai chiều
 - Mạng hai chiều là một mạng các phần tử của nó là mạng một chiều, giống như một bảng gồm có dòng và cột, được đánh dấu vị trí là các chỉ số bao gồm chỉ số dòng và chỉ số cột trong đó.

3. Mảng nhiều chiều

- Khai báo mảng hai chiều

`type arrayName[rows][columns];`

- rows: số hàng
- columns: số cột

- Ví dụ:

Khai báo mảng số nguyên 3 hàng 4 cột:

`int a[3][4]`

num [t] [i]

	0	1	2	3
0	1	2	3	4
1	5	6	7	8
2	9	10	11	12

3. Mảng nhiều chiều

- Khai báo mảng hai chiều

Cú pháp:

```
type arrayName[][columns] = { {value1,value2,...,valueN},  
{value1,value2,...,valueN}, {...}, {value1,value2,...,valueN}};
```

- Số phần tử của mỗi hàng phải bằng số cột
- Số hàng của khai báo mảng hai chiều để trống.
- Số hàng của mảng được xác định dựa vào số hàng trong phần khởi tạo. Giá trị các phần tử trong mỗi hàng được đặt trong cặp {}, các hàng phân cách nhau bằng một dấu phẩy.

3. Mảng nhiều chiều

- Khai báo mảng hai chiều

Ví dụ:

```
int a[][4] = {{ int a[][4] = {{1 2 3 4} {5 6 7 8} {9 10 11 12}};  
,2,3,4}, {5,6,7,8},{9,10,11,12}};
```

3. Mảng nhiều chiều

Ma trận, mảng 2 chiều:
Một nhóm các phần tử có cùng kiểu, chung tên.

Các phần tử được xác định bằng số dòng và số cột.

	Chỉ số cột				
	0	1	2	3	4
0	5	7	9	2	3
1	12	5	7	4	6
2	1	8	9	0	3
3	2	5	6	3	4

Chỉ số dòng

Ma trận m

$m[2][3]$

$m[i][j]$ is an integer

3. Mảng nhiều chiều

- Nhập mảng hai chiều

Ví dụ:

```
int main()
{
    int a[10][10];
    for (int i=0; i<2; i++)
        for (int j=0; j<3; j++)
        {
            printf("\nNhap a[%d][%d]: ",i,j);
            scanf("%d", &a[i][j]);
        }
}
```

```
int main()
{
    int a[10][10];
    for (int i=0; i<2; i++)
        for (int j=0; j<3; j++)
        {
            cout<<"n Nhap a["<<i<<"]["<<j<<":";
            cin>>a[i][j];
        }
}
```

3. Mảng nhiều chiều

■ Nhập mảng hai chiều

Ví dụ: nhập $a_{n \times m}$. Xây dựng hàm nhập
xây dựng hàm nhập

```
void NhapMaTran(int a[][100], int n, int m)
{
    for (int i=0; i<n; i++)
        for (int j=0; j<m; j++)
        {
            printf("\nNhap a[%d][%d]: ",i,j);
            scanf("%d", &a[i][j]);
        }
}
```

```
void nhap(int a[][4],int n,int m)
{
    for (int i=0; i<2; i++)
        for (int j=0; j<4; j++)
        {
            cout<<"n Nhap a["<<i<<"]["<<j<<":";
            cin>>a[i][j];
        }
}
```


3. Mảng nhiều chiều

■ Xuất mảng hai chiều

Ví dụ: xuất mảng $a_{n \times m}$

```
int main()
{
    int a[4][4]={{1,2,3,4},{5,6,7,8}};
    for(int i=0; i<2; i++)
    {
        for(int j=0; j<3; j++)
        {
            printf("%4d",a[i][j]);
        }
        printf("\n");
    }
}
```

```
int main()
{
    int a[4][4]={{1,2,3,4},{5,6,7,8}};
    for (int i=0; i<2; i++)
    {   for (int j=0; j<4; j++)
        {
            cout<<setw(3)<<a[i][j];
        }
        cout<<"\n";
    }
}
```

Hàm setw cần khai báo thư viện iomanip

3. Mảng nhiều chiều

- Xuất mảng hai chiều

Ví dụ: xuất mảng $a_{n \times m}$.

Xây dựng hàm xuất mảng 2 chiều:

```
void xuat(int a[][4],int n,int m)
{
    for (int i=0; i<n; i++)
    {
        for (int j=0; j<m; j++)
            cout<<setw(3)<<a[i][j];
        cout<<"\n";
    }
}
```

3. Mảng nhiều chiều

- Truy xuất mảng hai chiều

```
int main()
{
    int a[2][2]={{1,2},{3,4}};
    int sum=0;
    for(int i=0;i<2;i++)
        for(int j=0;j<2;j++)

        sum=sum+a[i][j];
    cout<<"\n sum:"<<sum;

}
```

3. Mảng nhiều chiều

```
#include <iostream.h>
#include <conio.h>
void main()
{
    int a[][4] = {{1,2,3,4},{5,6,7,8},{9,10,11,12},
                  {13,14,15,16}};
    int sum=0;
    for(int i=0 ; i<4 ; i++)
        for(int j=0 ; j<4 ; j++)
            if(i==j)
                sum += a[i][j];
    cout << "Tong duong cheo chinh la: " << sum;
}
```

Bài tập

1. Tìm phần tử lớn nhất trong mảng hai chiều cấp $n \times m$
2. Tổng các số nguyên tố trong mảng hai chiều cấp $n \times m$
3. Tìm dòng có tổng lớn nhất trong mảng hai chiều cấp $n \times m$
4. Tính tổng, hiệu, tích hai ma trận A, B cùng cấp n
5. Viết hàm thêm một hàng tại vị trí thứ k ($0 \leq k \leq n$) trong mảng 2 chiều $A[n \times m]$.
6. Viết hàm thêm một cột tại vị trí thứ k ($0 \leq k \leq m$) trong mảng 2 chiều $A[n \times m]$.

Bài tập

1. Viết hàm xóa một cột tại vị trí thứ k ($0 \leq k < n$) trong mảng 2 chiều $A[n \times m]$.
2. Viết hàm xóa một hàng tại vị trí thứ k ($0 \leq k < n$) trong mảng 2 chiều $A[n \times m]$.
3. Viết hàm hoán vị 2 hàng thứ k và l ($0 \leq k, l < n$) trong mảng 2 chiều $A[n \times m]$.
4. Viết hàm hoán vị 2 cột thứ k và l ($0 \leq k, l < m$) trong mảng 2 chiều $A[n \times m]$.
5. Viết hàm tính tổng các phần tử của hàng thứ k ($0 \leq k < n$) trong mảng 2 chiều $A[n \times m]$.

Bài tập

1. Viết hàm tính tổng các phần tử của cột thứ k ($0 \leq k < mm$) trong mảng 2 chiều $A[n \times m]$
2. Viết hàm sắp xếp mảng 2 chiều $A[n \times m]$ sao cho tổng của các dòng tăng dần từ trên xuống dưới.
3. Viết hàm sắp xếp mảng 2 chiều $A[n \times m]$ sao cho tổng của các cột tăng dần từ trái sang phải.
4. Cho ma trận vuông A $n \times n$. Viết các hàm sau:
 - a) Tính tổng các phần tử nằm trên đường chéo chính.
 - b) Tính tổng các phần tử nằm trên đường chéo phụ.
 - c) Tính tổng các phần tử nằm phía trên đường chéo chính.
 - d) Tính tổng các phần tử nằm phía trên đường chéo phụ.