

TIN HỌC ĐẠI CƯƠNG

BÀI 8: MẢNG

GV: Nguyễn Văn Đồng

BM Công nghệ phần mềm –Khoa Công nghệ thông tin

Email: nvdong@tlu.edu.vn

1. Mảng một chiều (dãy)
2. Mảng hai chiều (ma trận)

Mảng một chiều (dãy)

Mảng một chiều

- Là một dãy phần tử có cùng kiểu dữ liệu
- Khai báo mảng:
 kiểu–phần–tử tên–mảng[số–phần–tử];
- Ví dụ:

```
int a[9]; // Mảng a có 9 phần tử kiểu số  
          // nguyên, nhưng giá trị của  
          // các phần tử chưa xác định.
```

	0	1	2	3	4	5	6	7	8
Mảng a									

Phần tử mảng

- Truy nhập phần tử mảng bằng chỉ số (bắt đầu từ 0):
tên—mảng[chỉ—số]

- Ví dụ:

`int a[9];`

Các phần tử là $a[0], a[1], \dots, a[8]$

	0	1	2	3	4	5	6	7	8
Mảng a									

- Dùng phần tử mảng giống như một biến thông thường:

$a[0] = 1;$

$a[2] = 6;$

$\text{cout} \ll a[2] + 10;$

	0	1	2	3	4	5	6	7	8
Mảng a	1		6						

Khởi tạo mảng

- Sau khi khai báo mảng, các phần tử có giá trị không xác định
- Khởi tạo mảng tức là gán giá trị ban đầu cho các phần tử
- Khai báo kết hợp khởi tạo mảng:

`int a[5] = {4, -1, 7, 12, 8};`

(hoặc: `int a[] = {4, -1, 7, 12, 8};`)

- Phải đảm bảo kích thước mảng (5) không nhỏ hơn số giá trị dùng để khởi tạo (4, -1, 7, 12, 8); nếu lớn hơn thì các phần tử thừa ra sẽ được gán giá trị 0
 - Trong trường hợp không chỉ rõ kích thước mảng, kích thước mảng được xác định thông qua số giá trị khởi tạo
- Khai báo mảng, sau đó gán giá trị cho từng phần tử:

`double b[2];`

`b[0] = 1.5;`

`b[1] = 2.6;`

	0	1
b	1.5	2.6

Nhập và hiển thị một dãy số nguyên

```
#include <iostream>
#include <string>
using namespace std;
int main()
{
    int a[100]; // co 100 phan tu
    int n; // nhung chi lam viec voi n phan tu dau
    cout << "Nhap so phan tu: ";
    cin >> n;
    cout << "Nhap gia tri (nguyen) cho tung phan tu:\n";
    for (int i = 0; i < n; i++) {
        cout << "a[" << i << "] = ";
        cin >> a[i];
    }
    cout << "Day so vua nhap la:\n";
    for (int i = 0; i < n; i++) {
        cout << a[i] << " ";
    }
    return 0;
}
```

Bài tập xử lý dãy số (trên lớp)

1. Tính tổng và trung bình cộng của các phần tử của dãy số thực
2. Đếm số phần tử dương trong một dãy số thực và tính trung bình cộng của chúng
3. Tìm phần tử lớn nhất và vị trí đầu tiên của nó trong dãy số thực
4. Sắp xếp dãy số thực theo chiều tăng dần

Mảng hai chiều (ma trận)

Mảng hai chiều

- Là một tập phần tử có cùng kiểu dữ liệu, được tổ chức thành một ma trận (bảng) gồm các hàng và cột

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & \dots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \dots & a_{1n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m0} & a_{m2} & a_{m2} & \dots & a_{mn} \end{bmatrix}$$

- Khai báo:
kiểu—phần—tử tên—mảng[số—hàng][số—cột];
- Ví dụ:
`int a[3][6];` // Ma trận a có 3 hàng và 6 cột,
// gồm 18 phần tử kiểu số nguyên

Khai báo mảng hai chiều

- Sau khi khai báo, các phần tử có giá trị không xác định

`int a[3][6]`

		0	1	2	3	4	5
a	0						
	1						
	2						

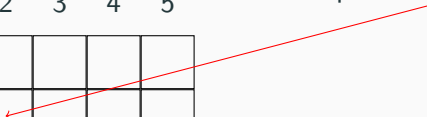
Phần tử mảng hai chiều (1)

- Truy nhập phần tử mảng hai chiều bằng chỉ số hàng và chỉ số cột (đều bắt đầu từ 0):

tên—mảng[chỉ—số—hàng][chỉ—số—cột]

	0	1	2	3	4	5
0						
a 1						
2						

phần tử a[1][2]



Phần tử mảng hai chiều (2)

- Dùng phần tử mảng hai chiều giống như một biến thông thường

```
a[0][1] = 8;  
a[1][2] = 3;  
a[1][5] = 2;  
cout << a[1][5] * 3; // sẽ in ra 6
```

	0	1	2	3	4	5
0		8				
1			3			2
2						

Nhập và hiển thị ma trận (1)

```
#include <iostream>
using namespace std;
int main()
{
    int a[100][100]; // Co 100 hang va 100 cot nhưng chi
    int m, n; // lam viec voi m hang va n cot dau tien.
    cout << "Nhap so hang: ";
    cin >> m;
    cout << "Nhap so cot: ";
    cin >> n;
    cout << "Nhap vao cac phan tu (nguyen):" << endl;
    for (int i = 0; i < m; i++) // i la chi so hang
        for (int j = 0; j < n; j++) // j la chi so cot
        {
            cout << "a[" << i << "][" << j << "] = ";
            cin >> a[i][j];
        }
}
```

Nhập và hiển thị ma trận (2)

```
// ... Tiếp tục từ slide trước
// Hiển thị ma trận
cout << "Ma trận mà bạn vừa nhập là:" << endl;
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
        cout << a[i][j] << " ";
    cout << endl; // in xong một hàng thì xuống dòng
}
return 0;
} // dấu ngoặc móc kết thúc hàm main
```

Bài tập xử lý ma trận (trên lớp)

1. Đếm số phần tử chẵn của một ma trận và tính trung bình cộng của chúng (các phần tử của ma trận có kiểu nguyên)
2. Tính tổng của các phần tử trên đường chéo chính của ma trận vuông
3. Kiểm tra một ma trận vuông có phải là ma trận đơn vị hay không (tức là các phần tử trên đường chéo chính bằng 1, trong khi các phần tử còn lại bằng 0)

Questions?