Chương 4

Kỹ thuật xử lý mảng

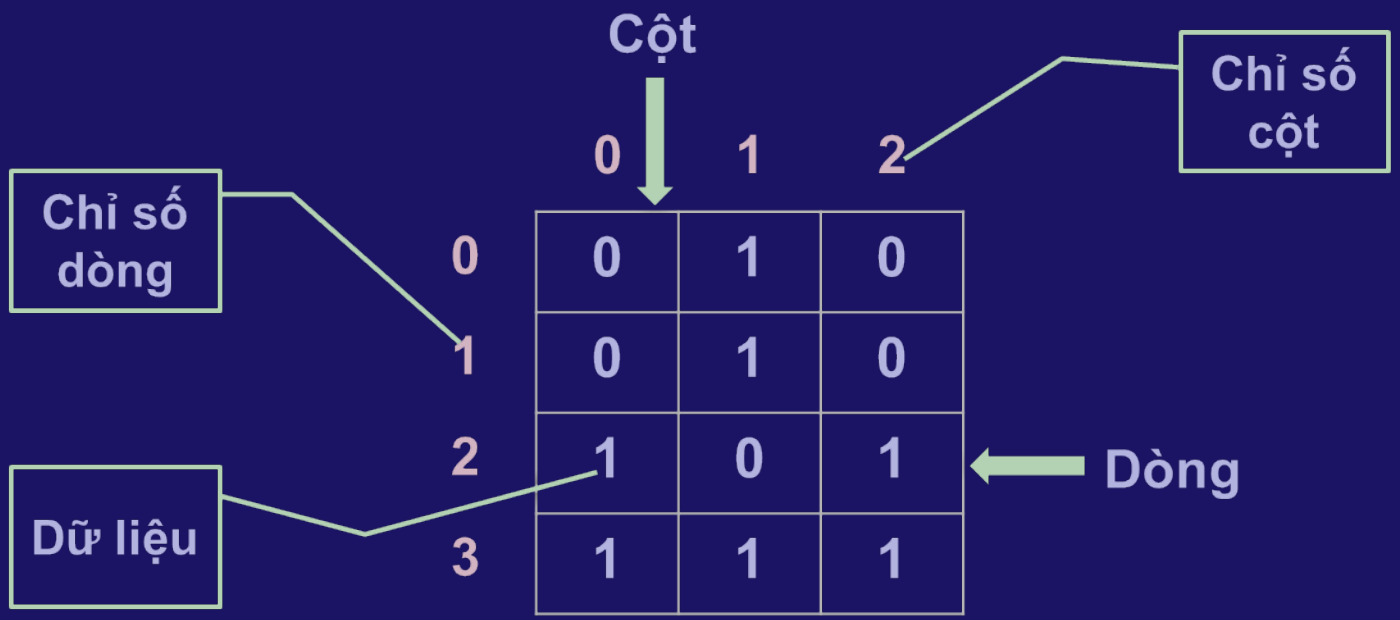
Giảng viên: Nguyễn Chiến Thắng
Email: thangnc.haui@gmail.com

Nội dung

1. Kỹ thuật xử lý mảng một chiều
2. Kỹ thuật xử lý mảng 2 chiều

4.4.1. Khái niệm mảng hai chiều

- Mảng đa chiều đơn giản nhất và thường được dùng nhất là mảng hai chiều gồm các dòng và các cột.



VD: Mảng hai chiều gồm 4 dòng và 3 cột.

Khái niệm mảng hai chiều (tt)

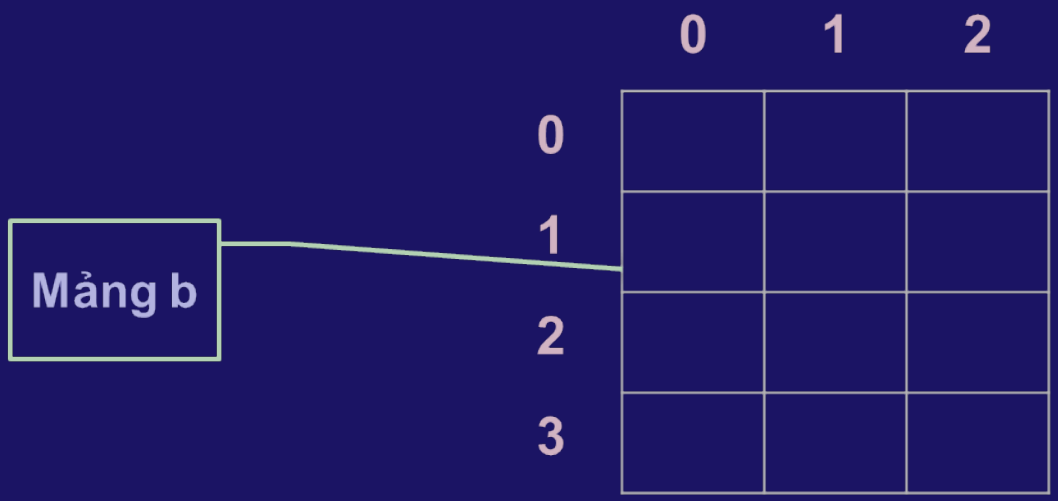
- Khai báo mảng hai chiều:

```
<Kiểu_Dữ_Liệu> <tên_mảng><[M][N]>;
```

Trong đó **M** là số dòng, **N** là số cột của mảng.

- Ví dụ:

```
int b[4][3];
```



Khái niệm mảng hai chiều (tt)

- Về logic, một mảng hai chiều giống như một **BẢNG THÀNH TÍCH** của các vận động viên cử tạ, gồm các dòng và các cột.
- Bảng thành tích của 4 vận động viên cử tạ, mỗi vận động viên cử 3 lần.

	Lần 1	Lần 2	Lần 3
Can	134kg	136kg	140kg
Bình	134kg	135kg	137kg
Dũng	135kg	137kg	143kg
Sỹ	123kg	135kg	136kg

Khái niệm mảng hai chiều (tt)

- Mảng hai chiều lưu trữ **BẢNG THÀNH TÍCH** của các vận động viên cử tạ.

	0	1	2
0	134	136	140
1	134	135	137
2	135	137	143
3	123	135	136

4.4.2. Khởi tạo mảng hai chiều

- Khởi tạo mảng trong lệnh khai báo.

```
int b[4][3] =    {
        { 134,136,140 },
        { 134,135,137 },
        { 135,137,143 },
        { 123,135,136 }
};
```

- Kết quả sau lệnh khởi tạo trên như sau:

```
b[0][0]=134; b[0][1]=136; b[0][2]=140;
b[1][0]=134; b[1][1]=135; b[1][2]=137;
b[2][0]=135; b[2][1]=137; b[2][2]=143;
b[3][0]=123; b[3][1]=135; b[3][2]=136;
```

Khởi tạo mảng hai chiều (tt)

- Nhập mảng từ bàn phím:
  - ✓ Nhập theo từng dòng.
  - ✓ Sử dụng hai vòng lặp lồng nhau.

```
//Nhập mảng hai chiều m dòng, n cột dữ liệu
void nhapMang(int b[4][3], int m, int n){
   for (int i = 0; i < m; i++){
      cout<<"Nhap dong thu "<<(i+1)<<endl;
      for (int j = 0; j < n; j++){
         cout<<"\tb["<<i<<"]["<<j<<"] = ";
         cin>>b[i][j];
      }
   }
}
```

4.4.3. Xử lý mảng hai chiều

- Hiển thị mảng ra màn hình.
  - ✓ Hiển thị theo dạng bảng.
  - ✓ Sử dụng 2 vòng lặp lồng nhau.

```
//Hiển thị mảng hai chiều m dòng, n cột dữ liệu
void hienThiMang(int b[4][3], int m, int n){
   for (int i = 0; i < m; i++){
      for (int j = 0; j < n; j++){
         cout<<"\t"<<b[i][j];
      }
      cout<<endl;
   }
}
```

Xử lý mảng hai chiều (tt)

1) Cài đặt chương trình quản lý các vận động viên cử tạ thi đấu trong trận chung kết, gồm m (m <= 7) vận động viên, mỗi vận động viên cử tạ n lần (n <= 3).
   - ✓ Nhập bảng thành tích của các vận động viên.
   - ✓ Hiển thị bảng thành tích lên màn hình.
   - ✓ Vận động viên nào có thành tích cử tạ cao nhất trong một lần cử tạ.
   - ✓ Vận động viên nào đoạt chức vô địch (có tổng 3 lần cử tạ tốt nhất).

Xử lý mảng hai chiều (tt)

2) Chương trình xử lý ma trận:

- ✓ Nhập ma trận vuông cấp n (1 ≤ n ≤ 10, n nhập từ bàn phím), mỗi phần tử là một số thực.
- ✓ Hiển thị ma trận ra màn hình.
- ✓ Tính và in ra màn hình tổng các phần tử trên đường chéo chính của ma trận.
- ✓ Tính và in ra màn hình tổng của các phần tử trên hàng chẵn, cột lẻ của ma trận.
- ✓ Cho biết ma trận có phải là ma trận tam giác trên hay không?

Xử lý mảng hai chiều (tt)

3) Chương trình xử lý ma trận:

- ✓ Tạo một ma trận xoắn ốc cấp mxn (1 ≤ m, n ≤ 20, m, n nhập từ bàn phím).
- ✓ Hiển thị ma trận ra màn hình.

	0	1	2	3	4	5
0	0	1	2	3	4	5
1	17	18	19	20	21	6
2	16	27	28	29	22	7
3	15	26	25	24	23	8
4	14	13	12	11	10	9

Thank you!