

# TIN ĐẠI CƯƠNG

## Bài 8: MẢNG MỘT CHIỀU, HAI CHIỀU

**Trần Thị Ngân**

Bộ môn Công nghệ phần mềm, Khoa CNTT

Trường đại học Thủy Lợi

# Nội dung chính

1. Khái niệm mảng một chiều
2. Khai báo mảng
3. Sử dụng mảng
4. Bài tập
5. Mảng 2 chiều

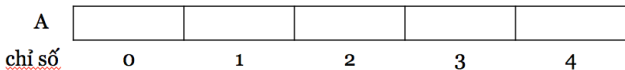
## 1. Khái niệm mảng một chiều

- Ví dụ: Nhập dữ liệu điểm môn Tin học đại cương cho 120 sinh viên lớp N03. Đếm và hiển thị ra màn hình các điểm thi trên 8.
- Sử dụng 120 biến phân biệt  $d_1, d_2, \dots, d_{120}$  để lưu trữ và xử lý điểm thi có các nhược điểm
  - Việc quản lí các biến không dễ dàng
  - Khó sử dụng cấu trúc lặp
  - Không tổng quát, khó nâng cấp chương trình

→ Giải pháp: sử dụng kiểu dữ liệu mảng

## Mảng

- Mảng là một dãy hữu hạn các phần tử có cùng kiểu dữ liệu và chiếm một vùng liên tục trong bộ nhớ
- Các phần tử của mảng được truy cập thông qua chỉ số của chúng (index)
- Ví dụ : Mảng A gồm 5 giá trị nguyên kiểu int.  
Các phần tử được đánh số từ 0 đến 4.



## 2. Khai báo mảng

- Cú pháp :

<kiểu dữ liệu> <tên mảng> [<số phần tử>];

- Ví dụ :

- `int A[5];` //mảng A có 5 phần tử dạng int
- `float B[10];` //mảng B có 10 phần tử dạng float với chỉ số từ 0 đến 9

- <số phần tử> là một hằng

## Khai báo mảng

- Có thể gán giá trị ban đầu cho các phần tử của mảng khi khai báo mảng bằng cách đặt các giá trị trong hai dấu { }
- Ví dụ :
  - `int A[5] = {100, 34, 23, 213, 500 };`
  - `int A[ ] = {100, 34, 23, 213, 500 } ;` //không chỉ định cỡ của mảng → khai báo mảng với cỡ đủ để chứa các giá trị khởi tạo

A	100	34	23	213	500
<u>chỉ số</u>	0	1	2	3	4

- `int A[5] = {100, 34, 23 } ;` //điền các giá trị này từ đầu mảng, điền phần còn lại với giá trị 0

A	100	34	23	0	0
<u>chỉ số</u>	0	1	2	3	4

### 3. Sử dụng mảng

- Sử dụng chỉ số đặt trong cặp dấu [ ] để truy cập đến từng phần tử trong mảng : <tên mảng>[<chỉ số>]

int A[5] = {100, 34, 23, 213, 500 } ;

→ A[0] = 100, A[1] = 34, A[2] = 23, A[3] = 213, A[4] = 500

- Có thể thao tác với các phần tử của mảng như thao tác với một biến thông thường

- A[0] -= 9; → A[0] = 91

- A[3] \*= 2; → A[3] = 426

- Chỉ số có thể là một biến nguyên hoặc một **biểu thức nguyên**

- size = 5; → A[size-1] = 500 //phần tử cuối cùng của mảng

- A[size/2] = A[2] = 23

## Chú ý

- Kích thước mảng phải cố định (mảng tĩnh)
- Mảng không thể thay đổi kích thước trong quá trình chương trình chạy
- Có thể dùng hằng có tên để chỉ định kích thước mảng

```
#include <iostream>
using namespace std;
int main()
{
    int kich_thuoc = 10;
    int a[kich_thuoc];
    return 0;
}
```

Trình biên dịch báo lỗi vì kich\_thuoc là biến có thể thay đổi giá trị

```
#include <iostream>
using namespace std;
int main()
{
    const int kich_thuoc = 10;
    int a[kich_thuoc];
    return 0;
}
```

Trình biên dịch không báo lỗi vì kich\_thuoc là hằng. Mảng a có 10 phần tử.



# Chú ý

- Chỉ số được đánh số từ 0 đến size-1.
- C++ "cho phép" vượt ra khỏi miền, trình biên dịch không phát hiện ra lỗi này. Trong trường hợp này

Kết quả không đoán trước được

Lập trình viên phải tự kiểm soát miền của chỉ số

- Lỗi hay gặp :

```
float diemthi[120];
```

```
diemthi[120] = 8.5; //chỉ số 120 nằm ở ngoài miền
```

## Một số thao tác cơ bản với mảng

- Nhập dữ liệu cho mảng

```
int A[5];  
for (int i=0; i<5; i++)  
{  
    cout << "Nhập phần tử thứ " << i << endl;  
    cin >> A[i];  
}
```

- In các giá trị của mảng ra màn hình

```
cout << "Day so vua nhap la \n";  
for (int i=0; i<5; i++) cout << A[i] << "  ";
```

## Một số bài toán với mảng

- Tính tổng các phần tử của mảng
- Đếm số các phần tử dương
- Tìm phần tử lớn nhất/nhỏ nhất và vị trí của chúng trong mảng
- Sắp xếp mảng theo thứ tự tăng dần/giảm dần

## Tìm phần tử lớn nhất của dãy

```
#include <iostream>
using namespace std;
int main()
{
    const int kichthuoc = 5;
    float A[kichthuoc], maxA;
    int i, k;
    //nhap day so
    for (i=0; i<kichthuoc; i++)
    {
        cout << "Nhap so thuc thu " << i << endl;
        cin >> A[i];
    }
    //tim max
    maxA = A[0]; k = 0;
    for (i=1; i<kichthuoc; i++)
        if (A[i] > maxA)
        {
            maxA = A[i];
            k = i;
        }
    cout << "So lon nhat la " << maxA << " tai vi tri " << k << endl;
    return 0;
}
```

Trong trường hợp mảng có nhiều số bằng nhau và lớn nhất thì các vị trí được hiển thị như thế nào?

# Sắp xếp dãy số theo thứ tự tăng

```
#include <iostream>
using namespace std;
int main()
{
    const int kichthuoc = 5;
    float A[kichthuoc], tg;
    int i, j;
    //nhap day so
    for (i=0; i<kichthuoc; i++)
    {
        cout << "Nhap so thuc thu " << i << endl;
        cin >> A[i];
    }
    //sap xep tang
    for (i=0; i<kichthuoc; i++)
        for (j=i+1; j<kichthuoc; j++)
            if (A[i] > A[j])
            {
                tg = A[i]; A[i] = A[j]; A[j] = tg;
            }
    //in ra day sau khi sap xep
    cout << "Day so sau khi sap xep:\n";
    for (i=0; i<kichthuoc; i++) cout << A[i] << "    ";
    return 0;
}
```

## 4. Bài tập

Bài 1: Nhập vào một dãy 10 số thực. Hiển thị dãy số đó trên màn hình.

Bài 2: Nhập vào một dãy 10 số nguyên và một số nguyên  $n$ .  
Đếm và chỉ ra vị trí các phần tử của dãy có giá trị

- Bằng  $n$ .
- Nhỏ hơn  $n$ .
- Là ước số của  $n$ .

Bài 3: Nhập vào một dãy 10 số thực. Tìm phần tử nhỏ nhất và vị trí của nó trong dãy số.

## 5. Mảng hai chiều

### 5.1. Khái niệm mảng hai chiều

- Mảng hai chiều là sự mở rộng của mảng một chiều. Về bản chất, nó là một danh sách các mảng một chiều.
- Các phần tử của mảng được truy cập thông qua hai chỉ số của chúng
- Ví dụ : Mảng A gồm  $3 \times 5$  phần tử được biểu diễn như sau

0					
1					
2					
	0	1	2	3	4

↓  
 $A[1][2]$

## 5.2. Khai báo mảng 2 chiều

- Cú pháp :

<kiểu dữ liệu> <tên mảng> [<số hàng>] [<số cột>];

Ví dụ :

```
int A[3[5];    //mảng A có 3*5 phần tử dạng int
```

- Truy cập đến các phần tử trong mảng :

<tên mảng> [<chỉ số hàng>] [<chỉ số cột>]

Ví dụ :  $A[1][2] = 8;$



## Khai báo mảng 2 chiều (tiếp)

- Có thể gán giá trị ban đầu cho các phần tử của mảng khi khai báo mảng

Ví dụ :

```
int A[3][5] = {  
    {0, 1, 2, 3, 4} ,           //hàng thứ nhất  
    {5, 6, 7, 8, 9} ,           //hàng thứ hai  
    {10, 11, 12, 13, 14}       //hàng thứ ba  
};
```

Hoặc:

```
int A[3][5] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14};
```

Hai cách làm trên là tương đương

## 5.4. Sử dụng mảng 2 chiều

- △ Tương tự như mảng một chiều, chỉ khác là có hai chỉ số đặt trong hai cặp dấu [ ]
- △ Nhắc lại :
  - △ kích thước mảng phải cố định
  - △ chỉ số được đánh số từ 0
  - △ lập trình viên phải tự kiểm soát miền của chỉ số vì trình biên dịch không báo lỗi khi chỉ số vượt ra khỏi miền

## 5.5. Một số thao tác cơ bản

- Nhập dữ liệu cho

```
int A[3][5];
for (int i=0; i<3; i++)
    for(int j=0; j<5; j++)
    {
        cout << "Nhap a[" << i << "][" << j << "]: ";
        cin >> A[i][j];
    }
```

- In các giá trị của mảng ra màn hình

```
cout << "Ma tran vua nhap la \n";
for (int i=0; i<3; i++)
{
    for(int j=0; j<5; j++) cout << A[i][j] << "  ";
    cout << endl;
}
```

## 5.5. Một số bài toán với mảng hai chiều

- Tương tự như với mảng một chiều, có các bài toán thực hiện tính toán trên toàn bộ mảng :
  - Tính tổng các phần tử của mảng
  - Đếm số các phần tử dương
  - Tìm phần tử lớn nhất/nhỏ nhất và vị trí của chúng trong mảng
- Các bài toán quy về mảng một chiều: tính toán trên một hàng, một cột hay trên đường chéo của mảng hai chiều

## 5. 6. Bài tập

**Bài 1:** Nhập vào một ma trận vuông các số thực. Tính trung bình cộng của các phần tử trên đường chéo chính của ma trận.

**Bài 2:** Nhập vào một ma trận các số thực. Tìm phần tử nhỏ nhất và vị trí của nó trong ma trận.