

# KỸ THUẬT LẬP TRÌNH

## Chương 1: Mảng nhiều chiều

# Mục tiêu

---

► Sau khi học xong chương này, người học có thể:

1

Biết cách sử dụng mảng 2 chiều: khai báo, khởi tạo, truy xuất

2

Các thao tác cơ bản với mảng 2 chiều: nhập xuất, tính toán, tìm kiếm, ...

3

Truyền mảng 2 chiều vào hàm

# Nội dung

---



## Mảng 2 chiều

1

Giới thiệu mảng 2 chiều

2

Khai báo, khởi tạo

3

Thao tác nhập/ xuất mảng 2 chiều

4

Tìm kiếm, tính toán với mảng 2 chiều

5

Truyền mảng 2 chiều cho hàm

- Bài toán: lưu trữ điểm số 13 môn học của học sinh phổ thông. Biết rằng lớp học đó có 40 học sinh.

	Môn 1	Môn 2	Môn 3	Môn 4	.....	Môn 12	Môn 13
Học sinh 1	<b>5</b>	<b>7.2</b>	<b>7.1</b>	<b>4.5</b>	...	<b>5.9</b>	<b>8.1</b>
Học sinh 2	<b>10</b>	<b>3</b>	<b>7.8</b>	<b>6.7</b>	...	<b>4.9</b>	<b>7.4</b>
Học sinh 3	<b>8</b>	<b>7</b>	<b>6</b>	<b>6.5</b>	...	<b>8.1</b>	<b>9.2</b>
...	...	...	...	...	...	...	...
Học sinh 39	<b>7.3</b>	<b>5.2</b>	<b>9</b>	<b>6</b>	...	<b>7.4</b>	<b>6.8</b>
Học sinh 40	<b>7.3</b>	<b>5.5</b>	<b>7.5</b>	<b>8</b>	...	<b>9.3</b>	<b>10</b>

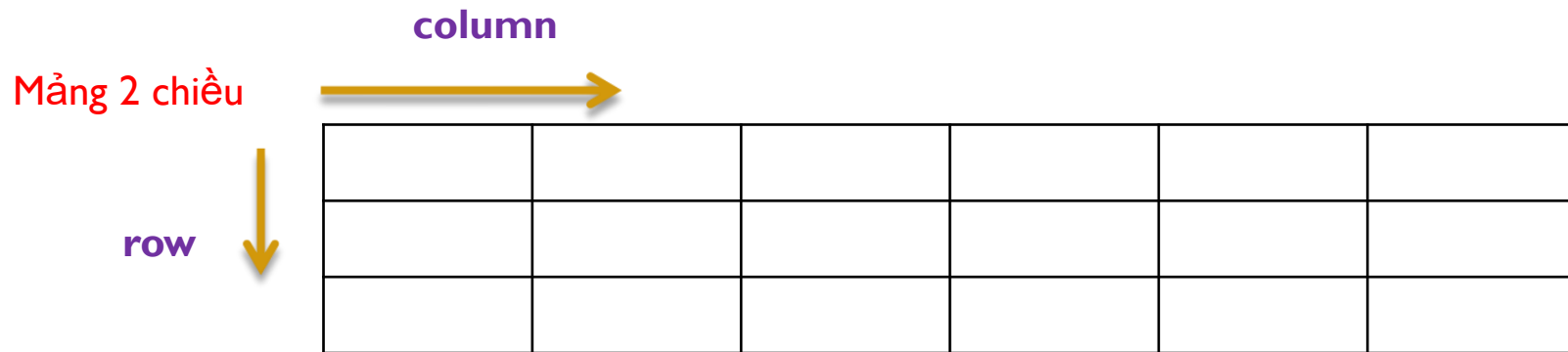
- **Mảng** (array) là một tập hợp nhiều phần tử có cùng kiểu dữ liệu, được lưu trữ tại các vị trí liên tục trong bộ nhớ.

**Mảng 1 chiều**

--	--	--	--	--	--	--

**Mảng 2 chiều**


- ▶ **Mảng 2 chiều (two – dimensional array)** lưu dữ liệu theo dạng bảng (table) dưới dạng gồm nhiều dòng (row) và nhiều cột (column).



- **Mảng 2 chiều (two – dimensional array)** có tên là **a** gồm **4 cột** và **3 dòng**

	Cột 0	Cột 1	Cột 2	Cột 3
Dòng 0	<b>a</b> <b>[0][0]</b>	<b>a</b> <b>[0][1]</b>	<b>a</b> <b>[0][2]</b>	<b>a</b> <b>[0][3]</b>
Dòng 1	<b>a</b> <b>[1][0]</b>	<b>a</b> <b>[1][1]</b>	<b>a</b> <b>[1][2]</b>	<b>a</b> <b>[1][3]</b>
Dòng 2	<b>a</b> <b>[2][0]</b>	<b>a</b> <b>[2][1]</b>	<b>a</b> <b>[2][2]</b>	<b>a</b> <b>[2][3]</b>

- ▶ Khai báo trước khi sử dụng.
- ▶ Cú pháp khai báo mảng 2 chiều:

```
dataType arrayName [numberOfRows] [numberOfColumns] ;
```

Trong đó:

- `dataType`: kiểu dữ liệu các phần tử trong mảng
- `arrayName`: tên mảng
- `NumberOfRows`: tổng số dòng
- `NumberOfColumns`: tổng số cột



- ▶ Khai báo trước khi sử dụng.
- ▶ Cú pháp khai báo mảng 2 chiều:

```
dataType arrayName [numberOfRows] [numberOfColumns] ;
```

**Ví dụ**: khai báo một *mảng 2 chiều gồm 3 hàng và 4 cột* để lưu trữ *điểm 4 môn học* cho *3 học sinh*.

```
double diem[3][4];
```



Mô phỏng cách lưu trữ???

► Cú pháp khai báo và khởi tạo mảng 2 chiều:

```
dataType arrayName[numberOfRows][numberOfColumns]
                                = { {initialValues},
                                    {initialValues},
                                    ...,
                                    {initialValues}};
```

Trong đó:

- `dataType`: kiểu dữ liệu các phần tử trong mảng
- `arrayName`: tên mảng
- `NumberOfRows`: tổng số dòng
- `NumberOfColumns`: tổng số cột
- `initialvalues`: giá trị từng phần tử ở mỗi dòng, đặt trong cặp dấu ngoặc { } và mỗi giá trị cách nhau bằng dấu phẩy.

## ► Cú pháp khai báo và khởi tạo mảng 2 chiều:

### Ví dụ:

```
int a[3][2] = { {0, 0}, {0, 0}, {0, 0} };
```

```
int a[3][2] = { {0}, {0}, {0} };
```

```
int a[3][2] = { {0} };
```

### Mảng a

<b>0</b>	<b>0</b>
<b>0</b>	<b>0</b>
<b>0</b>	<b>0</b>

## ▶ Nhập dữ liệu vào mảng 2 chiều:

- Số lượng phần tử tối đa lưu trữ ?
  - Cần số dòng tối đa có thể lưu trữ.
  - Cần số cột tối đa có thể lưu trữ.

```
const int MAXROW = 10, MAXCOL = 5;
```

- Cần duyệt từng phần tử để nhận vào giá trị tương ứng.

### ▶ Nhập dữ liệu vào mảng 2 chiều:

- Khai báo và gán giá trị cho toàn bộ phần tử.

Ví dụ: khởi tạo giá trị 0 cho mảng 2 chiều 4 dòng 3 cột:

```
for (int i = 0; i < 4; i++)  
    for (int j = 0; j < 3; j++)  
        a[i][j] = 0;
```

- Cho phép người dùng nhập giá trị từng phần tử.
- Cho phép người dùng nhập giá trị các phần tử dưới dạng từng dòng.

- Cho phép người dùng nhập giá trị từng phần tử: **Dùng vòng lặp và câu lệnh nhập**

```
int a[MAXROW][MAXCOL] ;  
int r, c;  
cout << "Ban can nhap so dong la: ";  
cin >> r;  
cout << "Ban can nhap so cot la: ";  
cin >> c;  
for (int i = 0; i < r; i++)  
    for (int j = 0; j < c; j++)  
    {  
        cout << "Nhap gia tri a[" << i << "][" << j << "]: ";  
        cin >> a[i][j];  
    }
```

- ▶ Nhập dữ liệu vào mảng 2 chiều bằng cách nhập giá trị từng dòng: **Dùng vòng lặp và câu lệnh nhập**

```
int a[MAXROW][MAXCOL] ;
int r, c;
cout << "Ban can nhap so dong la: ";
cin >> r;
cout << "Ban can nhap so cot la: ";
cin >> c;
for (int i = 0; i < r; i++)
{
    cout << "Nhap " << c << " so nguyen cho dong thu "
    << i + 1 << ": ";
    for (int j = 0; j < c; j++)
        cin >> a[i][j];
}
```

- ▶ Xuất dữ liệu trong mảng 2 chiều: dùng vòng lặp và câu lệnh xuất

```
cout << "Mang đang lưu trữ là: " << endl;
for (int i = 0; i < r; i++)
{
    for (int j = 0; j < c; j++)
        cout << a[i][j] << "\t";
    cout << endl;
}
```



- ▶ Tính toán trong mảng hai chiều
- Tính tổng toàn bộ các giá trị lưu trữ trong mảng 2 chiều:

```
int tong = 0;
```

```
for (int i = 0; i < r; i++)
```

```
    for (int j = 0; j < c; j++)
```

```
        tong+= a[i][j];
```

```
cout << "Tong cac phan tu la: " << tong << endl;
```

## Bài tập

- ▶ Bài 1: Hãy viết chương trình:
  - ▶ Nhập vào số dòng và cột cần lưu trữ của một mảng 2 chiều lưu trữ toàn số nguyên (tối đa 20 dòng và 10 cột). Nếu sai hãy yêu cầu nhập lại cho đến khi thỏa điều kiện.
  - ▶ Nhập vào giá trị từng phần tử trong mảng.
  - ▶ Xuất lại mảng đã nhập.
  - ▶ Tính tích các phần tử tại 1 dòng nào đó do người dùng yêu cầu.

## Bài tập

- ▶ Bài 2: Hãy viết chương trình dùng mảng 2 chiều xuất ra giá trị dãy số có dạng như sau:

1    2    3    4    5

6    7    8    9    10

11   12   13   14   15

16   17   18   19   20

## Bài tập

- ▶ Bài 3: Hãy viết chương trình dùng câu lệnh while để nhập và xuất giá trị một mảng số nguyên gồm 5 hàng và 3 cột.

- ▶ Tìm kiếm trong mảng hai chiều: dùng câu lệnh lặp để duyệt và tìm kiếm bằng cách so sánh giá trị đang duyệt với giá trị cần tìm.

```
int x;
bool flag = false;
cout << "Ban can tim: ";
cin >> x;
for (int i = 0; i < r; i++)
    for (int j = 0; j < c; j++)
        if (a[i][j] == x)
        {
            flag = true;
            break;
        }
if (flag)
    cout << x << " co trong mang" << endl;
else
    cout << x << " khong co trong mang" << endl;
```

## Bài tập

- ▶ Bài 4: Viết chương trình cho phép nhập vào một mảng số nguyên tối đa 10 dòng và 5 cột. Sau đó tìm xem một giá trị x nào đó (do người dùng nhập) có trong mảng hay không? Bao nhiêu lần? Vị trí đầu tiên xuất hiện là ở đâu?

---

▶ Truyền mảng hai chiều vào hàm

- Mảng 2 chiều truyền vào hàm theo tham chiếu (nên truyền tham số hằng cho việc xuất mảng), nên truyền kèm theo kích thước mỗi chiều.

```
void nhap (int a[3][4]) ;
```

```
void nhap (int a[MAX][MAX], int r, int c) ;
```

- Có thể bỏ qua tham số dòng tối đa lưu trữ (nhưng phải là tham số thứ 2 khi truyền cho hàm).

```
void nhap (int a[][4], int r) ;
```

## ▶ Truyền mảng hai chiều vào hàm

### ◦ Hàm nhập mảng 2 chiều:

```
void nhap (int a[MAXROW][MAXCOL], int r, int c)
{
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
        {
            cout << "Nhập giá trị cho phần tử a[" << i
            << "][" << j << "]: ";
            cin >> a[i][j];
            cout << endl;
        }
}
```



## ▶ Truyền mảng hai chiều vào hàm

### ○ Hàm xuất mảng 2 chiều:

```
void xuat (int a[MAXROW][MAXCOL], int r, int c)
{
    cout << "Mang da nhap la: " << endl;
    for (int i = 0; i < r; i++)
    {
        for (int j = 0; j < c; j++)
            cout << a[i][j] << "\t";
        cout << endl;
    }
}
```

▶ Truyền mảng hai chiều vào hàm

- Hàm tính tổng các giá trị mảng 2 chiều:

```
int tinh tong (int a[MAXROW][MAXCOL], int r, int c)
{
    int tong = 0;
    for (int i = 0; i < r; i++)
        for (int j = 0; j < c; j++)
            tong += a[i][j];
    return tong;
}
```

► Gọi hàm thực thi:

```
int main()
{
    int a[MAXROW][MAXCOL] ;
    int r, c;
    cout << "Ban can nhap so dong la: ";
    cin >> r;
    cout << endl;
    cout << "Ban can nhap so cot la: ";
    cin >> c;
    cout << endl;
    nhap(a, r, c);
    xuat(a, r, c);
    cout << endl << "Tong cac phan tu cua mang la: " <<
    tinhhtong(a, r, c) << endl;
}
```

# Nhược điểm của mảng kiểu C

---

- ▶ Kích thước của mảng không thay đổi được trong thời gian thực thi.

---

# Q & A