

# MỤC TIÊU

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**

- **Giới thiệu mảng 2 chiều**
- Khai báo, khởi tạo
- Thao tác nhập/xuất mảng 2 chiều
- Tìm kiếm, tính toán với mảng 2 chiều
- Truyền mảng 2 chiều cho hàm



# CHƯƠNG 1: **MẢNG NHIỀU CHIỀU**

# GIỚI THIỆU

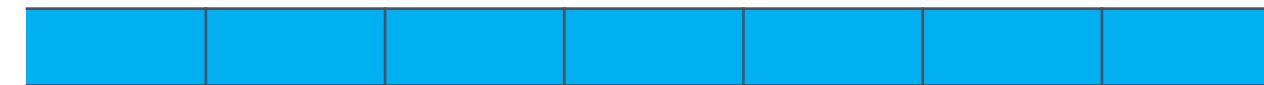
**Bài toán:** lưu trữ điểm số 13 môn học của học sinh phổ thông. Biết rằng lớp học đó có 40 học sinh

|             | Môn 1 | Môn 2 | Môn 3 | Môn 4 | ... | Môn 12 | Môn 13 |
|-------------|-------|-------|-------|-------|-----|--------|--------|
| Học sinh 1  | 5     | 7.2   | 7.1   | 4.5   | ... | 5.3    | 8.1    |
| Học sinh 2  | 10    | 4     | 6.2   | 7.1   | ... | 8.1    | 9      |
| Học sinh 3  | 8     | 6     | 7     | 6.5   | ... | 7.2    | 9.2    |
| ...         | ...   | ...   | ...   | ...   | ... | ...    | ...    |
| Học sinh 39 | 7.3   | 5.2   | 9     | 6     | ... | 7.4    | 6.8    |
| Học sinh 40 | 7.1   | 5.5   | 7.9   | 10    | ... | 10     | 9.3    |

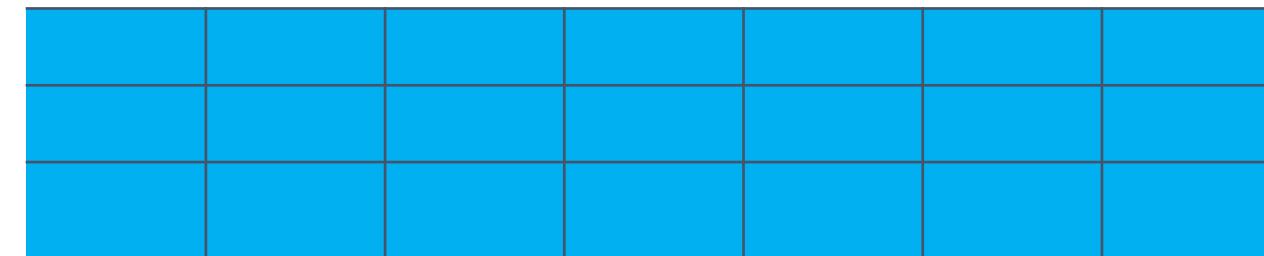
# GIỚI THIỆU

**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**



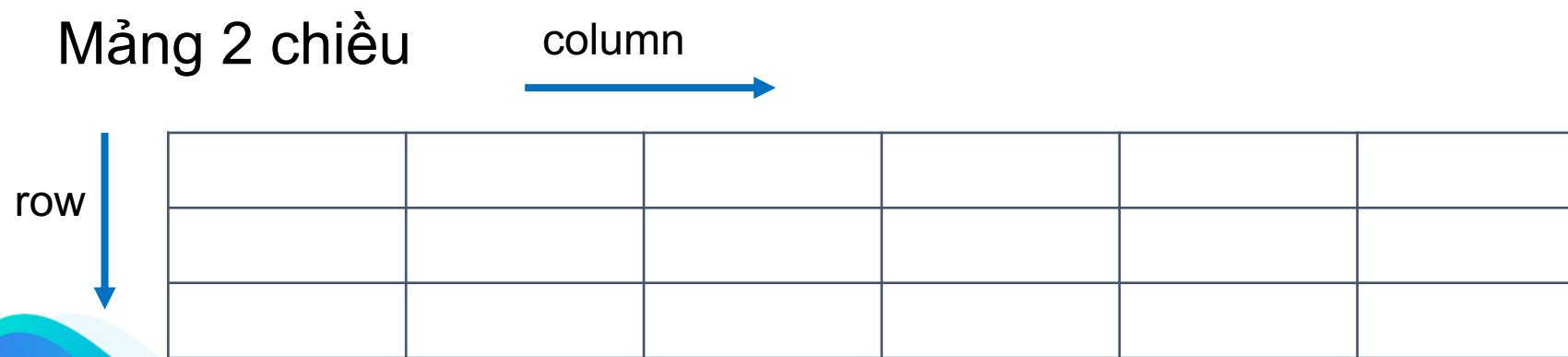
**Mảng 2 chiều**



# GIỚI THIỆU

**Mảng 2 chiều (two – dimensional array)** lưu trữ dữ liệu

theo dạng bảng (table) dưới dạng gồm nhiều dòng (row)  
và nhiều cột (column).



# GIỚI THIỆU

**Mảng 2 chiều (two – dimensional array)** có tên là a gồm 4 cột và 3 dòng -> tổng số phần tử là 12 phần tử

Ví dụ mảng 2 chiều:

|        | Cột 0   | Cột 1   | Cột 2   | Cột 3   |
|--------|---------|---------|---------|---------|
| Dòng 0 | a[0][0] | a[1][1] | a[1][1] | a[1][1] |
| Dòng 1 | a[1][0] | a[1][1] | a[1][1] | a[1][1] |
| Dòng 2 | a[2][0] | a[1][1] | a[1][1] | a[1][1] |

- Giới thiệu mảng 2 chiều
- Khai báo, khởi tạo
- Thao tác nhập/xuất mảng 2 chiều
- Tìm kiếm, tính toán với mảng 2 chiều
- Truyền mảng 2 chiều cho hàm



# CHƯƠNG 1: MẢNG NHIỀU CHIỀU

# KHAI BÁO, KHỞI TẠO

Lưu ý:

- ❑ Cần khai báo mảng trước khi sử dụng.
- ❑ Cú pháp khai báo như sau:

**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, KHỞI TẠO

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];
```

*Một mảng 2 chiều không có giá trị mặc định. Do đó ta buộc phải gán giá trị cho các phần tử trong mảng*

# KHAI BÁO, KHỞI TẠO

Ví dụ: khai báo và khởi tạo mảng gồm 3 dòng và 2 cột với các giá trị là 0.

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

Mảng a:

|   |   |
|---|---|
| 0 | 0 |
| 0 | 0 |
| 0 | 0 |

- Giới thiệu mảng 2 chiều
- Khai báo, khởi tạo
- **Thao tác nhập/xuất mảng 2 chiều**
- Tìm kiếm, tính toán với mảng 2 chiều
- Truyền mảng 2 chiều cho hàm



# CHƯƠNG 1: **MẢNG NHIỀU CHIỀU**

# THAO TÁC NHẬP/XUẤT

**Một số yêu cầu khi nhập dữ liệu vào mảng 2 chiều:**

Số lượng phần tử tối đa mà mảng có thể lưu trữ?

- Cần số dòng tối đa **cần** lưu trữ
- Cần số cột tối đa **cần** lưu trữ

**Ví dụ:** Const int MAXROW = 10, MAXCOL = 5

# THAO TÁC NHẬP/XUẤT

Nhập dữ liệu vào mảng 2 chiều: mảng cần duyệt qua **từng phần tử** để gán giá trị tương ứng.

Khai báo và gán giá trị cho toàn bộ phần tử trong một mảng:

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

```
const int MAXROW = 10, MAXCOL = 5;  
  
int a[MAXROW][MAXCOL]; ←  
for (int i = 0; i < 4; i++) {  
    for (int j = 0; j < 3; j++) {  
        a[i][j] = 0;  
    }  
}
```

Khai báo mảng chứa số phần tử tối đa

# THAO TÁC NHẬP/XUẤT

Viết chương trình 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];
    }
}
```

# THAO TÁC NHẬP/XUẤT

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

```
//giả sử rằng, rows và columns đã được khai báo trước
cout << "Mang dang luu tru la: " << endl;
for (int i = 0; i < rows; i++) {
    for (int j = 0; j < columns; j++) {
        cout << a[i][j] << "\t";
    }
    cout << endl;
}
```

- Giới thiệu mảng 2 chiều
- Khai báo, khởi tạo
- Thao tác nhập/xuất mảng 2 chiều
- Tìm kiếm, tính toán với mảng 2 chiều
- Truyền mảng 2 chiều cho hàm



# CHƯƠNG 1: MẢNG NHIỀU CHIỀU

# TÌM KIẾM, TÍNH TOÁN

Tính toán trong mảng 2 chiều:

Ví dụ: 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;
```

tong cần được gán giá trị bằng 0 khi khai báo để tránh việc chương trình gán giá trị 'rác' dẫn tới sai kết quả

# TÌM KIẾM, TÍNH TOÁN

**Tìm kiếm trong mảng hai chiều:**

**Ví dụ:** 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;
```

- Giới thiệu mảng 2 chiều
- Khai báo, khởi tạo
- Thao tác nhập/xuất mảng 2 chiều
- Tìm kiếm, tính toán với mảng 2 chiều
- Truyền mảng 2 chiều cho hàm



# CHƯƠNG 1: MẢNG NHIỀU CHIỀU

# TRUYỀN MẢNG CHO HÀM

- ② 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), ví dụ:  

```
void nhap (int a [ ] [4], int r); // tối đa 4 dòng 4 cột.
```

# TRUYỀN MẢNG CHO HÀM

## Truyền mảng hai chiều vào hàm

- Viết 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 CHO HÀM

## Truyền mảng hai chiều vào hàm

- Viết 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 CHO HÀM

## Truyền mảng hai chiều vào hàm

- Viết 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;
}
```

# TRUYỀN MẢNG CHO HÀM

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: " <<  
    tinh tong(a, r, c) << endl;  
  
}
```



**THANK YOU**