

Bài 36: CÁC THAO TÁC TRÊN MẢNG 2 CHIỀU

Xem bài học trên website để ủng hộ Kteam: [Các thao tác trên Mảng 2 chiều](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn về [MẢNG HAI CHIỀU \(Two-dimensional arrays\)](#) trong C++.

Hôm nay, mình sẽ giới thiệu cho các bạn về **Các thao tác trên Mảng 2 chiều trong C++**.

Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về:

- [VÒNG LẶP FOR TRONG C++ \(For statements\)](#)
- [MẢNG MỘT CHIỀU \(Arrays\)](#)
- [MẢNG HAI CHIỀU \(Two-dimensional arrays\)](#)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Truyền mảng vào hàm (passing arrays to functions)
- Nhập và xuất mảng 2 chiều
- Tính tổng các phần tử trong mảng
- Tìm giá trị lớn nhất của mảng 2 chiều

Truyền mảng vào hàm (passing arrays to functions)

Lưu ý khi truyền mảng vào hàm:

Tham số kiểu mảng trong khai báo hàm **giống như khai báo biến mảng**.

```
void NhapMang(int a[100][100]);
```

- Tham số kiểu mảng truyền cho hàm chính là **địa chỉ của phần tử đầu tiên** của mảng.
 - Có thể **bỏ số lượng phần tử dòng** hoặc **sử dụng con trỏ**.
 - Mảng **có thể thay đổi nội dung** sau khi thực hiện hàm.

```
void NhapMang(int a[][100]);  
void NhapMang(int (*a)[100]);
```

- Số lượng phần tử thực sự truyền qua biến khác.

```
void NhapMang(int a[100][100], int m, int n);  
void NhapMang(int a[][100], int m, int n);  
void NhapMang(int (*a)[100], int m, int n);
```

Tất cả các ví dụ bên dưới sẽ sử dụng phương pháp truyền mảng vào hàm.

Nhập và xuất mảng 2 chiều

Bên dưới là ví dụ về nhập, xuất dữ liệu cho mảng 1 chiều:

```
#include <iostream>  
#include <cstdlib> // for srand() and rand()  
#include <ctime> // for time()  
using namespace std;  
  
// định nghĩa số phần tử mảng
```

```
#define ROW 100
#define COL 100

// khai báo prototype
void nhapMang(int mt[][COL], int &m, int &n);
void xuatMang(int mt[][COL], int m, int n);

int main()
{
    int mt[ROW][COL]; // mảng myArray có MAX phần tử
    int nRow, nCol; // số phần tử dòng và cột được sử dụng

    // nhập xuất mảng tự động
    nhapMang(mt, nRow, nCol);
    xuatMang(mt, nRow, nCol);

    return 0;
}

// hàm nhập mảng
void nhapMang(int mt[][COL], int &m, int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

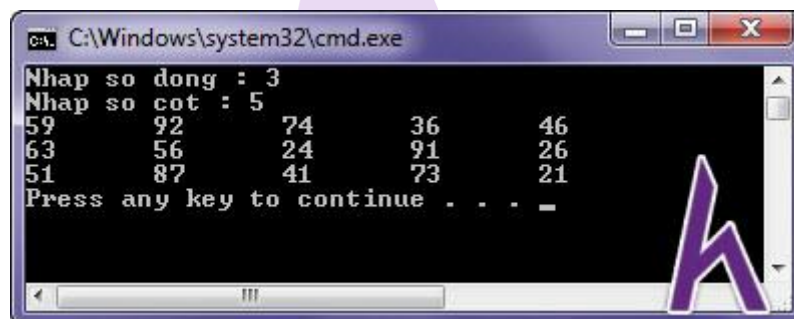
    cout << "Nhap so dong : ";
    cin >> m;
    cout << "Nhap so cot : ";
    cin >> n;

    // khởi tạo mảng ngẫu nhiên từ [0][0] đến [m-1][n-1].
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            mt[i][j] = rand() % 100;
        }
    }
}

// hàm xuất mảng
void xuatMang(int mt[][COL], int m, int n)
{

```

```
// xuất từng phần tử cho mảng
for (int i = 0; i < m; i++)
{
    for (int j = 0; j < n; j++)
    {
        cout << mt[i][j] << " ";
    }
    cout << endl;
}
}
```

Output:

Tính tổng các phần tử trong mảng

Yêu cầu: cho trước ma trận **mt**, kích thước **mxn**. Tính tổng các phần tử trên dòng **d**, cột **c**.

Ý tưởng: duyệt ma trận và cộng dồn các phần tử có tọa độ (dòng, cột) thỏa yêu cầu.

```
#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
using namespace std;

// định nghĩa số phần tử mảng
#define ROW 100
#define COL 100

// khai báo prototype
```

```
void nhapMang(int mt[][COL], int &m, int &n);
void xuatMang(int mt[][COL], int m, int n);
int tongDong(int mt[][COL], int n, int d);
int tongCot(int mt[][COL], int m, int c);

int main()
{
    int mt[ROW][COL]; // mảng myArray có MAX phần tử
    int nRow, nCol; // số phần tử dòng và cột được sử dụng

    // nhập xuất mảng tự động
    nhapMang(mt, nRow, nCol);
    xuatMang(mt, nRow, nCol);

    // tính tổng dòng d, cột c
    int d;
    cout << "Nhap dong can tinh tong: ";
    cin >> d;
    cout << "Tong dong: " << tongDong(mt, nCol, d) << endl;

    // tính tổng cột c
    int c;
    cout << "Nhap dong can tinh tong: ";
    cin >> c;
    cout << "Tong dong: " << tongCot(mt, nRow, c) << endl;

    return 0;
}

// hàm nhập mảng
void nhapMang(int mt[][COL], int &m, int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhap so dong : ";
    cin >> m;
    cout << "Nhap so cot : ";
    cin >> n;

    // khởi tạo mảng ngẫu nhiên từ [0][0] đến [m-1][n-1].
    for (int i = 0; i < m; i++)
```

```
{
    for (int j = 0; j < n; j++)
    {
        mt[i][j] = rand() % 100;
    }
}

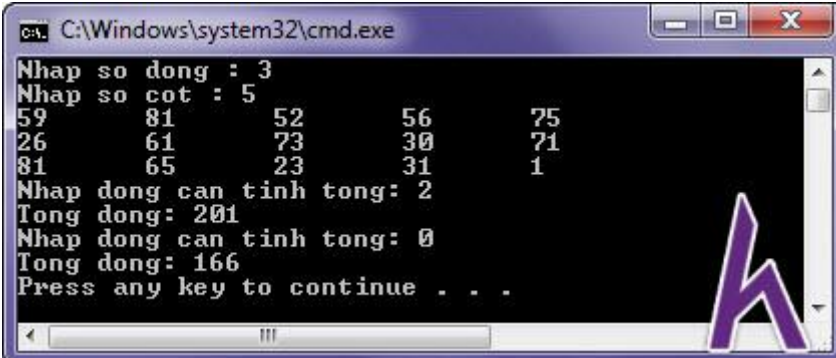
// hàm xuất mảng
void xuấtMang(int mt[][COL], int m, int n)
{
    // xuất từng phần tử cho mảng
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            cout << mt[i][j] << "t";
        }
        cout << endl;
    }
}

// tính tổng trên dòng d
int tongDong(int mt[][COL], int n, int d)
{
    int sum(0);
    for (int j = 0; j < n; j++) // Duyệt các cột
        sum += mt[d][j];

    return sum;
}

// tính tổng trên cột c
int tongCot(int mt[][COL], int m, int c)
{
    int sum(0);
    for (int i = 0; i < m; i++) // Duyệt các dòng
        sum += mt[i][c];

    return sum;
}
```

Output:


```

C:\Windows\system32\cmd.exe
Nhap so dong : 3
Nhap so cot : 5
59      81      52      56      75
26      61      73      30      71
81      65      23      31      1
Nhap dong can tinh tong: 2
Tong dong: 201
Nhap dong can tinh tong: 0
Tong dong: 166
Press any key to continue . . .

```

Tìm giá trị lớn nhất của mảng 2 chiều

Yêu cầu: Cho trước ma trận **mt**, kích thước **m**x**n**. Tìm giá trị lớn nhất trong ma trận **mt** (gọi là **max**)

Ý tưởng:

- Giả sử giá trị **max** hiện tại là giá trị phần tử đầu tiên **mt[0][0]**
- Lần lượt kiểm tra các phần tử còn lại để cập nhật **max**.

```

#include <iostream>
#include <cstdlib> // for srand() and rand()
#include <ctime> // for time()
using namespace std;

// định nghĩa số phần tử mảng
#define ROW 100
#define COL 100

// khai báo prototype
void nhapMang(int mt[][COL], int &m, int &n);
void xuatMang(int mt[][COL], int m, int n);
int timMax(int mt[][COL], int m, int n);

int main()
{

```

```
int mt[ROW][COL]; // mảng myArray có MAX phần tử
int nRow, nCol; // số phần tử dòng và cột được sử dụng

// nhập xuất mảng tự động
nhapMang(mt, nRow, nCol);
xuatMang(mt, nRow, nCol);

// tìm max
int max = timMax(mt, nRow, nCol);
cout << "Max = " << max << endl;

return 0;
}

// hàm nhập mảng
void nhapMang(int mt[][COL], int &m, int &n)
{
    // khởi tạo số ngẫu nhiên
    srand(time(NULL));

    cout << "Nhap so dong : ";
    cin >> m;
    cout << "Nhap so cot : ";
    cin >> n;

    // khởi tạo mảng ngẫu nhiên từ [0][0] đến [m-1][n-1].
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            mt[i][j] = rand() % 100;
        }
    }
}

// hàm xuất mảng
void xuatMang(int mt[][COL], int m, int n)
{
    // xuất từng phần tử cho mảng
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
```



```
        cout << mt[i][j] << "\\t";  
    }  
    cout << endl;  
}  
}  
  
// tìm giá trị max của mảng 2 chiều  
int timMax(int mt[][COL], int m, int n)  
{  
    int i, j, max;  
    max = mt[0][0];  
    for (i = 0; i < m; i++)  
        for (j = 0; j < n; j++)  
            if (mt[i][j] > max)  
                max = mt[i][j];  
    return max;  
}
```

Output:



Kết luận

Qua bài học này, bạn đã biết được [Các thao tác trên Mảng 2 chiều](#) trong C++. Còn rất nhiều thao tác khác trên mảng 2 chiều, trong phạm vi bài học không thể đề cập hết được, các bạn hãy tự mình tìm hiểu và bình luận bên dưới để chia sẻ cho mọi người nhé.

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn [MẢNG KÝ TỰ \(C-type strings\)](#) trong C++.

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.

