Bài 35: MANG 2 CHIỀU TRONG C++ (TWO-DIMENSIONAL ARRAYS)

Xem bài học trên website để ủng hộ Kteam: Mảng 2 chiều trong C++ (Two-dimensional arrays)

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 <u>How Kteam</u> nhé!

Dẫn nhập

Ở bài học trước, mình đã chia sẻ cho các bạn về <u>CÁC THAO TÁC TRÊN MẢNG</u> MỘT CHIỀU TRONG C++.

Hôm nay, mình sẽ giới thiệu cho các bạn về một dạng khác của kiểu dữ liệu mảng, đó là Mảng 2 chiều (Two-dimensional arrays) trong C++.

Nội dung

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

- VÒNG LĂP FOR TRONG C++ (For statements)
- MÅNG MỘT CHIỀU TRONG C++ (Arrays)

Trong bài ta sẽ cùng tìm hiểu các vấn đề:

- Mảng 2 chiều là gì?
- Khai báo và khởi tạo mảng 1 chiều
- Xuất các phần tử mảng 2 chiều
- Nhập các phần tử mảng 2 chiều



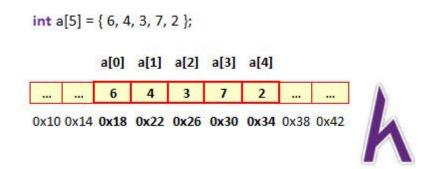
Mảng 2 chiều là gì?

Trong bài MẢNG MỘT CHIỀU TRONG C++ (Arrays), bạn đã biết được mảng là một kiểu dữ **liệu có cấu trúc** do người lập trình định nghĩa, biểu diễn **một dãy các biến có cùng kiểu**.

Ví dụ:

```
int a[5] = { 6, 4, 3, 7, 2 };
```

Hình bên dưới mô tả 1 mảng tên là **a** có kiểu **int** gồm **5 phần tử (đã khởi tạo)** nằm trong vùng nhớ RAM:



Các phần tử trong một mảng, có thể là **bất kỳ kiểu dữ liệu nào**, và nó cũng có thể là kiểu dữ liệu mảng. Nếu các phần tử trong mảng có kiểu dữ liệu mảng, nó được gọi là mảng đa chiều.

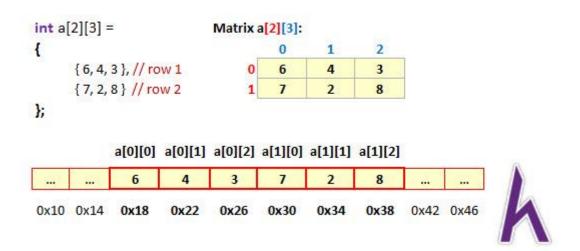
Một mảng của một mảng được gọi là **mảng đa chiều** (multidimensional array).

Ví dụ mảng 2 chiều:

```
int a[2][3] =
{
      { 6, 4, 3 }, // row 1
      { 7, 2, 8 } // row 2
};
```



Hình bên dưới mô tả mảng 2 chiều tên là a có kiểu int gồm 2 phần tử (đã khởi tạo), mỗi phần tử là mảng 1 chiều 3 phần tử (thường được gọi là 2 dòng 3 cột):



Dù mảng **a** là **2 chiều** hay **nhiều chiều**, nhưng bản chất khi lưu vào bộ nhớ RAM, nó vẫn là **1 vùng nhớ gồm những ô nhớ liên tiếp nhau**.

Khai báo và khởi tạo mảng 2 chiều Khai báo mảng 2 chiều Cú pháp:

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

Lưu ý:

Phải xác định <số phần tử dòng> và <số phần tử cột> cụ thể (hằng số) khi khai báo.



- Nên sử dụng chỉ thị tiền xử lý #define để định nghĩa <số phần tử> mảng.
- Tổng số phần tử = <số phần tử dòng> * <số phần tử cột>.
- Bộ nhớ sử dụng = <tổng số phần tử > * sizeof(<kiểu cơ sở >).

Ví dụ:

```
int a[10][20], b[10][20];
int c[5][10];
int d[10][20];
```

Chú ý: Khi một mảng chưa khởi tạo, các phần tử của mảng sẽ mang giá trị rác.

Khởi tạo giá trị cho mảng 2 chiều

Cách 1: Khởi tạo giá trị cho mọi phần tử của mảng

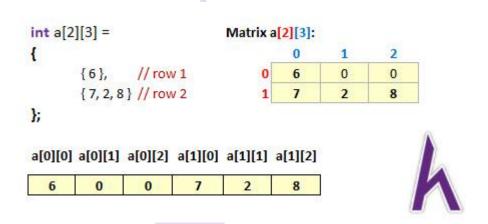
```
int a[2][3] =
{
    { 6, 4, 3 }, // row 1
    { 7, 2, 8 } // row 2
};
```

a[0][0] a[0][1] a[0][2] a[1][0] a[1][1] a[1][2]





Cách 2: Khởi tạo giá trị cho một số phần tử đầu mảng



Cách 3: Khởi tạo giá trị 0 cho mọi phần tử của mảng

Cách 4: Tự động xác định số lượng phần tử

```
int a[][3] =
{
      { 6, 4, 3 }, // row 1
      { 7, 2, 8 } // row 2
};
```



Không giống như mảng 1 chiều, **mảng 2 chiều không cho phép** khai báo bên dưới:

Cách 5: Sử dụng khởi tạo đồng nhất (uniform initialization) trong C++11 (tương tự như những cách trên, nhưng bỏ đi dấu bằng "=")



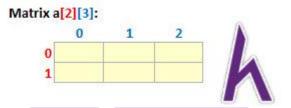
Xuất các phần tử mảng 2 chiều

Để truy xuất giá trị của phần tử trong mảng, ta sử dụng cú pháp:

<tên biến mảng>[<chỉ số dòng thứ i>][<chỉ số cột thứ j>];

Ví dụ: Cho mảng như sau:

int a[2][3] // 2 dòng, 3 cột



Các truy xuất:

- **Hợp lệ:** a[0][0], a[0][1], ..., a[1][1], a[1][2]
- Không hợp lệ: a[-1][0], a[0][-1], a[2][3], ... => cho kết quả không như mong muốn (có thể gây chết chương trình).

Chú ý: khi truy xuất một phần tử mảng, luôn đảm bảo chỉ số của phần tử đó là hợp lệ trong phạm vi của mảng.

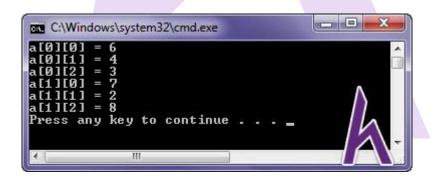
Ví dụ chương trình khởi tạo và xuất các phần tử mảng:

#include <iostream>
using namespace std;



```
#define ROW 2 // Max row
#define COL 3 // Max column
int main()
        // khởi tạo mảng 2 dòng 3 cột
        int arr[ROW][COL]
        {
                 { 6, 4, 3 }, // row 1
                 {7, 2, 8} // row 2
        };
        // xuất giá trị các phần tử mảng
        for (int i = 0; i < ROW; i++)
                 for (int j = 0; j < COL; j++)
                          cout << "a[" << i << "][" << j <<"] = " << arr[i][j] <<
endl;
                 }
        }
         return 0;
```

Output:



Chương trình trên sử dụng **2 vòng lặp for**, 1 vòng lặp chạy từ **0** đến **ROW – 1** để truy cập vào từng dòng, trên mỗi dòng sẽ có 1 vòng lặp chạy từ 0 đến **COL – 1** để truy cập vào từng cột.



Nhập các phần tử mảng 2 chiều

Để **gán giá trị cho phần tử** trong mảng, ta sử dụng **cú pháp**:

```
<ten biến mảng>[<chỉ số dòng thứ i>][<chỉ số cột thứ j>] = <giá
trị>;
```

Ví dụ các phép gán hợp lệ:

```
int a[2][3];

a[0][0] = 6;

a[0][1] = 4;

a[0][2] = 3;

a[1][0] = 7;

a[1][1] = 2;

a[1][2] = 8;
```

Ví dụ các phép gán KHÔNG hợp lệ (gây chết chương trình):

```
int a[2][3];
a[-1][0] = 6;
a[0][-1] = 4;
a[2][3] = 3;
```

Chú ý: khi truy cập một phần tử mảng, luôn đảm bảo chỉ số của phần tử đó là hợp lệ trong phạm vi của mảng.

Ví dụ chương trình yêu cầu nhập dữ liệu cho mảng 2 chiều, sau đó xuất ra màn hình:

```
#include <iostream>
using namespace std;

#define ROW 100 // Max row
```



```
#define COL 100 // Max column
int main()
        // khởi tạo mảng 2 dòng 3 cột
         int arr[ROW][COL];
         int row, col;
         // nhập số dòng, cột
         cout << "Nhap so dong: ";
         cin >> row;
         cout << "Nhap so cot: ";
         cin >> col;
         // nhập mảng 2 chiều
         cout << "Nhap mang:" << endl;
         for (int i = 0; i < row; i++)
                  for (int j = 0; j < col; j++)
                  {
                           cout << "a[" << i << "][" << j << "] = ";
                           cin >> arr[i][j];
                  }
         }
         // xuất mảng 2 chiều
         for (int i = 0; i < row; i++)
         {
                  for (int j = 0; j < col; j++)
                          cout << arr[i][j] << " ";
                  cout << endl;
         }
         return 0;
```

Output:



```
Nhap so dong: 2
Nhap so cot: 3
Nhap mang:
a[0][0] = 5
a[0][1] = 7
a[0][2] = 2
a[1][0] = 4
a[1][1] = 8
a[1][2] = 1
5 7 2
4 8 1
Press any key to continue . . .
```

Ví dụ chương trình phát sinh dữ liệu cho mảng 2 chiều, sau đó xuất ra màn hình:

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
#define ROW 100 // Max row
#define COL 100 // Max column
int main()
        // khởi tạo mảng 2 dòng 3 cột
        int arr[ROW][COL];
        int row, col;
        // khởi tạo số ngẫu nhiên
        srand(time(NULL));
        // nhập số dòng, cột
        cout << "Nhap so dong: ";
        cin >> row;
        cout << "Nhap so cot: ";
        cin >> col;
        // nhập mảng 2 chiều
        for (int i = 0; i < row; i++)
                 for (int j = 0; j < col; j++)
```



Output:

```
C:\Windows\system32\cmd.exe

Nhap so dong: 2
Nhap so cot: 3
3645 5299 20380
10865 21492 21333
Press any key to continue . . .
```

Kết luận

Qua bài học này, bạn đã biết được khái niệm và cách sử Mảng 2 chiều (Two-dimensional arrays) trong C++. Mảng 2 chiều được sử dụng khá phổ biến trong lập trình, nó được ứng dụng vào các bài toán đồ thị, hoặc các trò chơi như game caro, cờ vua, ...

Trong bài tiếp theo, mình sẽ giới thiệu cho các bạn <u>CÁC THAO TÁC TRÊN</u> <u>MẢNG 2 CHIỀU</u> 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**".

