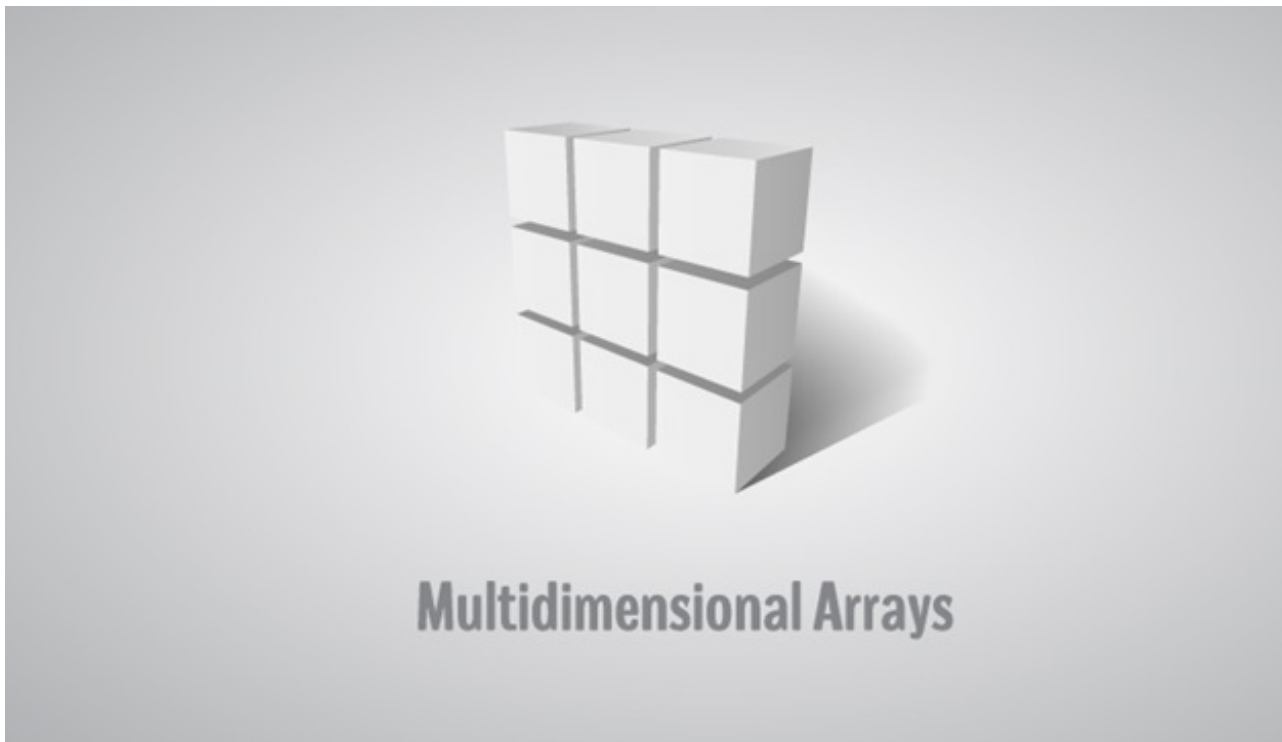


Mảng nhiều chiều

Trong bài này, bạn sẽ học về mảng nhiều chiều (multidimensional arrays) trong C++. Cụ thể hơn, cách khai báo, truy xuất và sử dụng chúng hiệu quả trong chương trình của bạn.



Trong C++, bạn có thể tạo một mảng của một mảng, được gọi là mảng nhiều chiều. Ví dụ:

```
int x[3][4];
```

Ở đây, x là một mảng hai chiều, nó có thể chứa tối đa 12 phần tử.

Bạn có thể coi mảng này giống như bảng với 3 hàng và mỗi hàng có 4 cột như sau:

	Cột 1	Cột 2	Cột 3	Cột 4
Hàng 1	x[0][0]	x[0][1]	x[0][2]	x[0][3]
Hàng 2	x[1][0]	x[1][1]	x[1][2]	x[1][3]
Hàng 3	x[2][0]	x[2][1]	x[2][2]	x[2][3]

Mảng ba chiều cũng tương tự như vậy. Ví dụ:

```
float x[2][4][3];
```

Mảng `x` này có thể chứa tối đa 24 phần tử. Bạn có thể xem ví dụ này giống như: Mỗi 2 phần tử có thể chứa 4 phần tử khác, qua đó ta có 8 phần tử và mỗi phần tử trong số 8 phần tử này lại có thể chứa 3 phần tử khác. Vì vậy số lượng phần tử tối đa cho mảng này là 24.

Khởi tạo mảng nhiều chiều

Bạn có thể khởi tạo một mảng nhiều chiều theo nhiều cách khác nhau.

Khởi tạo mảng hai chiều

```
int test[2][3] = {2, 4, -5, 9, 0, 9};
```

Cách tốt hơn để khởi tạo mảng này với cùng phần tử như trên.

```
int test[2][3] = { {2, 4, 5}, {9, 0, 0}};
```

Khởi tạo mảng ba chiều

```
int test[2][3][4] = {3, 4, 2, 3, 0, -3, 9, 11, 23, 12, 23, 2, 13, 4, 56, 3, 5, 9, 3, 5, 5, 1, 4, 9};
```

Cách tốt hơn để khởi tạo mảng này với cùng phần tử như trên

```
int test[2][3][4] = {
    {3, 4, 2, 3}, {0, -3, 9, 11}, {23, 12, 23, 2},
    {13, 4, 56, 3}, {5, 9, 3, 5}, {3, 1, 4, 9}
};
```

Ví dụ 1: Mảng hai chiều

Chương trình C++ giúp hiển thị toàn bộ phần tử của một mảng hai chiều đã khởi tạo.

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

int main()
{
    int test[3][2] =
    {
        {2, -5},
        {4, 0},
        {9, 1}
    };

    // Truy xuất mảng hai chiều sử dụng
    // vòng lặp for lồng nhau.
    for(int i = 0; i < 3; ++i)
    {
        for(int j = 0; j < 2; ++j)
        {
            cout<< "test[" << i << "]" << j << "] = " << test[i][j] << endl;
        }
    }

    return 0;
}
```

```
}
```

Đầu ra

```
test[0][0] = 2
test[0][1] = -5
test[1][0] = 4
test[1][1] = 0
test[2][0] = 9
test[2][1] = 1
```

Ví dụ 2: Mảng hai chiều

Chương trình C++ giúp lưu nhiệt độ của hai thành phố khác nhau trong 1 tuần và in ra.

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

const int CITY = 2;
const int WEEK = 7;

int main()
{
    int temperature[CITY][WEEK];

    cout << "Enter all temperature for a week of first city and then second city. \n";

    // Chèn dữ liệu vào mảng nhiệt độ
    for (int i = 0; i < CITY; ++i)
    {
        for(int j = 0; j < WEEK; ++j)
        {
            cout << "City " << i + 1 << ", Day " << j + 1 << " : ";
            cin >> temperature[i][j];
        }
    }

    cout << "\n\nDisplaying Values:\n";

    // Truy xuất dữ liệu từ mảng nhiệt độ
    for (int i = 0; i < CITY; ++i)
    {
        for(int j = 0; j < WEEK; ++j)
        {
            cout << "City " << i + 1 << ", Day " << j + 1 << " = " << temperature[i][j] << endl;
        }
    }

    return 0;
}
```

Đầu ra

Enter all temperature for a week of first city and then second city.

City 1, Day 1 : 32

City 1, Day 2 : 33

City 1, Day 3 : 32

City 1, Day 4 : 34

City 1, Day 5 : 35

City 1, Day 6 : 36

City 1, Day 7 : 38

City 2, Day 1 : 23

City 2, Day 2 : 24

City 2, Day 3 : 26

City 2, Day 4 : 22

City 2, Day 5 : 29

City 2, Day 6 : 27

City 2, Day 7 : 23

Displaying Values:

City 1, Day 1 = 32

City 1, Day 2 = 33

City 1, Day 3 = 32

City 1, Day 4 = 34

City 1, Day 5 = 35

City 1, Day 6 = 36

City 1, Day 7 = 38

City 2, Day 1 = 23

City 2, Day 2 = 24

City 2, Day 3 = 26

City 2, Day 4 = 22

City 2, Day 5 = 29

City 2, Day 6 = 27

City 2, Day 7 = 23

Ví dụ 3: Mảng ba chiều

Chương trình C++ giúp lưu giá trị nhập vào bởi người dùng trong một mảng 3 chiều và in ra.

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

int main()
{
    // Mảng này có thể chứa 12 phần tử (2x3x2)
    int test[2][3][2];

    cout << "Enter 12 values: \n";

    // Chèn dữ liệu vào mảng
    // sử dụng 3 vòng lặp for lồng nhau.
    for(int i = 0; i < 2; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            for(int k = 0; k < 2; ++k )
            {
                cin >> test[i][j][k];
            }
        }
    }

    cout<<"\nDisplaying Value stored:"<<endl;

    // Hiển thị giá trị với chỉ số phù hợp.
    for(int i = 0; i < 2; ++i)
    {
        for (int j = 0; j < 3; ++j)
        {
            for(int k = 0; k < 2; ++k)
            {
                cout << "test[" << i << "][" << j << "][" << k << "] = " << test[i][j][k] << endl;
            }
        }
    }

    return 0;
}
```

Đầu ra

Enter 12 values:

1
2
3
4
5
6
7
8
9
10
11
12

Displaying Value stored:

test[0][0][0] = 1

test[0][0][1] = 2

test[0][1][0] = 3

test[0][1][1] = 4

test[0][2][0] = 5

test[0][2][1] = 6

test[1][0][0] = 7

test[1][0][1] = 8

test[1][1][0] = 9

test[1][1][1] = 10

test[1][2][0] = 11

test[1][2][1] = 12

Khi số lượng chiều tăng lên, độ phức tạp cũng tăng rất nhanh mặc dù khái niệm khá tương đồng.