**Bài tập**

Cho mảng 2 chiều các số nguyên arr với n hàng m cột được nhập từ bàn phím, bạn hãy viết chương trình cấp phát bộ nhớ động cho vùng nhớ chứa mảng này sau đó hiển thị ra màn hình tổng của các hàng trong mảng.

Ví dụ nếu bạn nhập n = 3, m = 3, arr = [[3, 5, 2], [6, 5, 9], [1, 7, 3]] như bên dưới:

3 3

3 5 2

6 5 9

1 7 3

Thì màn hình sẽ hiển thị ra:

10

20

11

Giải thích: tổng của các hàng trong mảng là 10, 20, 11.

**Lý thuyết**

Bài này có một số cách để làm, trước hết bạn cần hiểu rõ hơn về bản chất của mảng 2 chiều.

Mảng 2 chiều là mảng các mảng 1 chiều, và mảng 2 chiều cũng có tính chất giống mảng 1 chiều là các phần tử trong mảng được nằm trong các ô nhớ liên tiếp nhau. Để hiểu rõ hơn bạn hãy xem ví dụ sau:

#include<iostream>

using namespace std;

int main() {

char arr[4][4];

for (int i = 0; i < 4; i++) {

for (int j = 0; j < 4; j++) {

cout << (int)&arr[i][j] << endl;

}

}

return 0;

}

Kết quả khi chạy chương trình:

19920928

19920929

19920930

19920931

19920932

19920933

19920934

19920935

19920936

19920937

19920938

19920939

19920940

19920941

19920942

19920943

Có thể thấy địa chỉ của các phần tử trong mảng là liên tiếp nhau, vậy bản chất mảng 2 chiều được lưu trữ trên bộ nhớ như là mảng một chiều. Do đó, bạn có thể truy xuất tới các phần tử trong mảng 2 chiều giống như mảng 1 chiều. Ví dụ chương trình sau:

#include<iostream>

using namespace std;

int main() {

int arr[3][3] = { {1, 2, 3}, {4, 5, 6}, {7, 8, 9} };

int\* p = &arr[0][0];

for (int i = 0; i < 9; i++) {

cout << p[i] << " ";

}

return 0;

}

Kết quả khi chạy chương trình:

1 2 3 4 5 6 7 8 9

Có thể thấy mảng 2 chiều có thể sử dụng giống như 1 mảng một chiều. Do đó bài này bạn cũng có thể làm bằng cách cấp phát bộ nhớ động cho mảng 1 chiều và sử dụng mảng 1 chiều này như là mảng 2 chiều.

Vậy chương trình trên có thể làm bằng mảng 1 chiều giống như sau:

#include<iostream>

using namespace std;

int main() {

int n, m;

cin >> n >> m;

int\* arr = new int[n \* m];

for (int i = 0; i < n \* m; i++) {

cin >> arr[i];

}

int index = 0;

for (int i = 0; i < n; i++) {

int sum = 0;

for (int j = 0; j < m; j++) {

sum += arr[index];

index++;

}

cout << sum << endl;

}

delete[]arr;

return 0;

}

**Cách 2: cấp phát động cho mảng 2 chiều.**

Có thể thấy cách làm bên trên có 1 số hạn chế như bạn không thể truy xuất tới phần tử hàng i cột j của mảng arr bằng arr[i][j] (do arr là mảng 1 chiều) và code theo cách này cũng khá phức tạp.

Mảng 2 chiều được cấp phát động thực ra là mảng một chiều các con trỏ, và các con trỏ này trỏ tới phần tử đầu tiên của mảng được cấp phát động do đó cấp phát động cho mảng 2 chiều thực ra là cấp phát động cho nhiều mảng 1 chiều.

Cụ thể hơn là bạn sẽ dùng 1 con trỏ cấp 2 (con trỏ trỏ tới con trỏ), cấp phát cho nó n con trỏ cấp 1 (con trỏ cấp 1 là các con trỏ mà bạn vẫn hay dùng như int\* p; // p là con trỏ cấp 1) và với mỗi con trỏ cấp 1 lại cấp phát m phần tử. Cụ thế hơn nếu làm với cách cấp phát động cho mảng 2 chiều thì sẽ như sau:

#include<iostream>

using namespace std;

int main() {

int n, m;

cin >> n >> m;

// Khai báo con trỏ cấp 2 arr

int\*\* arr;

// Cấp phát vùng nhớ động cho mảng 1 chiều các con trỏ kiểu int\*

// Có thể hiểu arr là mảng n phần tử, mỗi phần tử là 1 con trỏ kiểu int\*

arr = new int\* [n];

for (int i = 0; i < n; i++) {

// Cấp phát động cho các n mảng 1 chiều

arr[i] = new int[m];

}

// Nhập dữ liệu cho các phần tử trong mảng

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cin >> arr[i][j];

}

}

// Tính tổng từng hàng và hiển thị ra màn hình

for (int i = 0; i < n; i++) {

int sum = 0;

for (int j = 0; j < m; j++) {

sum += arr[i][j];

}

cout << sum << endl;

}

// Giải phóng bộ nhớ

for (int i = 0; i < n; i++) {

// Giải phóng bộ nhớ con các mảng một chiều

delete[] arr[i];

}

// Giải phóng bộ nhớ cho mảng các con trỏ

delete[]arr;

return 0;

}

Lưu ý: Các phần tử của mảng 2 chiều được cấp phát động sẽ không nằm trên các ô nhớ liên tiếp. Xem ví dụ sau:

#include<iostream>

using namespace std;

int main() {

int n = 3, m = 3;

// Khai báo con trỏ cấp 2 arr

int\*\* arr;

// Cấp phát vùng nhớ động cho mảng 1 chiều các con trỏ kiểu int\*

// Có thể hiểu arr là mảng n phần tử, mỗi phần tử là 1 con trỏ kiểu int\*

arr = new int\* [n];

for (int i = 0; i < n; i++) {

// Cấp phát động cho các mảng 1 chiều

arr[i] = new int[m];

}

for (int i = 0; i < n; i++) {

for (int j = 0; j < m; j++) {

cout << (int)&arr[i][j] << endl;

}

}

// Giải phóng bộ nhớ

for (int i = 0; i < n; i++) {

// Giải phóng bộ nhớ con các mảng một chiều

delete[] arr[i];

}

// Giải phóng bộ nhớ con mảng các con trỏ

delete[]arr;

return 0;

}

Kết quả khi chạy chươn trình:

9477304

9477308

9477312

9477808

9477812

9477816

9477360

9477364

9477368

Đọc tới đây bạn đã biết cách cấp phát bộ nhớ động cho mảng 2 chiều, hãy quay lại phần bài tập và làm thử.

**Hướng dẫn**

Code mẫu đã được cung cấp ở phần lý thuyết.