https://www.includehelp.com/stl/initialize-2d-vector…

Initialize 2D vector in C++ in different ways

Web Clip

2) Initialize the vector with user defined size

We can **initialize the 2D vector** with user-defined size also. It's quite similar like creating a 2D dynamic array using *malloc()* or *new* operator. So say we want to initialize a 2D vector to rows n, and column m, then we need to initialize an n size 2D vector with elements of m size 1D vector. We can do that just like below, by default all the valued in the 2D array gets initialized as 0.

As we said earlier a 2D vector is a vector of a 1D vector. So for the upper use case, let's think exactly similarly as of 1D vector.

So the outer vector has size n(number of rows)

Let's define that,

```
vector<T> arr(n);
```

Now *T* is itself a 1D vector and has size m

Thus the element of the outer vector would *vector<int>(m)*

This combining,

```
vector<vector<int>> arr(n, vector<int>(m))
```

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    //n =no of rows
    //m =no of columns
    //both will be user defined
    int n, m;

    cout << "Enter number of rows, n\n";
    cin >> n;
    cout << "Enter number of columns, m\n";
    cin >> m;

    //2D vector initialized with user defined size
    vector<vector<int> > two_D_vector(n, vector<int>(m));

    //by default all values are 0
    //printing the 2D vector
    cout << "printing the 2D vector\n";
    for (auto it : two_D_vector) {
```

```
    for (auto it : two_D_vector) {
        //it is now an 1D vector
        for (auto ij : it) {
            cout << ij << " ";
        }
        cout << endl;
    }

    return 0;
}
```

Output:

```
Enter number of rows, n
6
Enter number of columns, m
3
printing the 2D vector
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
0 0 0
```