Code Overview

You're using **nested vectors** (2D vectors) to store a grid-like structure and modifying it using a function.

1. Headers and Namespace

#include<iostream>

#include<vector>

using namespace std;

- #include<iostream> → for input/output (cin, cout)
- #include<vector> → for using the C++ STL vector
- using namespace std; → so you don't have to write std:: again and again.

2. Function to Modify Vector

```
void change2Dvector(vector<vector<int>>& v) {
  v[0][0] = 10;
}
```

Explanation:

- The function takes a **reference** to a 2D vector (&v) to avoid making a **copy**.
- v[0][0] = 10; modifies the **first element** of the first row of the vector.
- If you remove the &, the vector passed will be copied, and changes will not affect the original vector in main().

3. Creating Vectors in main()

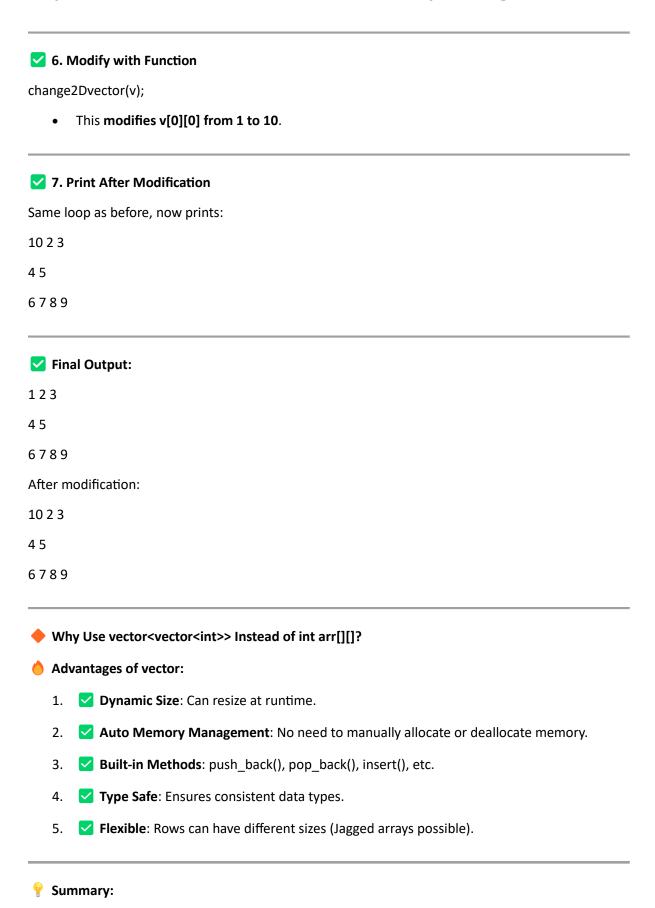
4. Pushing into 2D Vector

```
v.push_back(v1);
v.push_back(v2);
v.push_back(v3);
Now v becomes:
[
   [1, 2, 3],
   [4, 5],
   [6, 7, 8, 9]
]
```

5. Printing Before Modification

```
for(int i=0;i<v.size();i++){
   for(int j=0;j<v[i].size();j++){
      cout<<v[i][j]<<" ";
   }
   cout<<endl;
}

      Output:
1 2 3
4 5
6 7 8 9</pre>
```



Concept **Explanation**

vector<vector<int>> A dynamic 2D array

push_back() Adds elements

Pass by Reference (&) Lets the function change the original vector

v[i][j] Access element at row i, column j

When row sizes vary or memory flexibility is needed Use case

Let me know if you want this as a printable PDF note or explained with a diagram 🚣

