Goldmine2 in C++

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

int maxGold = 0;

void travel(vector<vector<int>>& arr, int i, int j,
vector<vector<bool>>& visited, vector<int>& bag) {
    if (i < 0 || j < 0 || i >= arr.size() || j >=
arr[0].size() || arr[i][j] == 0 || visited[i][j]) {
        return;
    }
    visited[i][j] = true;
    bag.push_back(arr[i][j]);
    travel(arr, i - 1, j, visited, bag);
    travel(arr, i, j + 1, visited, bag);
    travel(arr, i, j - 1, visited, bag);
    travel(arr, i + 1, j, visited, bag);
}

void getMaxGold(vector<vector<int>>& arr) {
    int rows = arr.size();
    int cols = arr[0].size();
    vector<vector<bool>> visited(rows,
vector<bool>(cols, false));

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (arr[i][j] != 0 && !visited[i][j]) {
                vector<int> bag;
                travel(arr, i, j, visited, bag);

                int sum = 0;
                for (int val : bag) {
                    sum += val;
                }
                if (sum > maxGold) {
                    maxGold = sum;
                }
            }
        }
    }
}

int main() {
    vector<vector<int>> arr = {
        {0, 1, 4, 2, 8, 2},
        {4, 3, 6, 5, 0, 4},
        {1, 2, 4, 1, 4, 6},
        {2, 0, 7, 3, 2, 2},
        {3, 1, 5, 9, 2, 4},
        {2, 7, 0, 8, 5, 1}
    };

    getMaxGold(arr);
    cout << maxGold << endl;

    return 0;
}
```

Step-by-Step Execution

Initial Setup:

- maxGold = 0
- visited initialized to false for all cells.
- Rows = 6, Cols = 6.

Outer Loop Iteration (i = 0, j = 0):

- Cell (0,0) is 0, skip it.

(i = 0, j = 1):

- Cell (0,1) is 1, not visited.
- Start travel function:
  - bag = [1]
  - Mark (0,1) as visited.
  - Explore neighboring cells:
    - (0,2): Add 4 to bag → bag = [1,4].
    - (1,2): Add 6 to bag → bag = [1,4,6].
    - Continue visiting valid cells → bag = [1,4,6,5,3,4,2].
  - Total sum of bag = 25.
- Update maxGold = 25.

(i = 0, j = 2, j = 3, ..., j = 5):

- All these cells are either visited or part of the same cluster.

Outer Loop Iteration (i = 1):

(i = 1, j = 0):

- Cell (1,0) is 4, not visited.
- Start travel function:
  - bag = [4]
  - Visit neighboring cells:
    - (2,0): Add 1 → bag = [4,1].
    - Continue visiting → bag = [4,1,2,3,7,9,5].
  - Total sum of bag = 31.
- Update maxGold = 31.

(i = 1, j = 1, ..., j = 5):

	• All cells are either visited or part of the same cluster.
	
	Outer Loop Iteration (i = 2 to i = 5):
	
	Continue similar logic for unvisited cells and new clusters.
	
	Cluster at (4,3) and `(5,3):
	
	• Explore the large gold cluster:
	  ○ bag = [9,5,8,7,2].
	  ○ Total sum = **120**.
	• Update maxGold = 120.
	
	Final Output:
	
	• **Maximum Gold Collected: 120**

Output:- 120	