Submission

ID	DATE	PROBLEM	STATUS	CPU	LANG
	TEST CASES				
7985128	05:59:19	Islands	✔ **Accepted**	0.00 s	C++
	✔✔				

Submission contains 1 file: download zip archive

FILENAME	FILESIZE	SHA-1 SUM	
islands3.cpp	1926 bytes	3df91d548022d0f561c2531e5fdfed54801c5b3f	download

Edit and resubmit this submission.

islands3.cpp

```
#include <bits/stdc++.h>
typedef long long ll;
//busqueda en profundidad
class Solution
{
public:
    ll i, j, k;
    void floodFill(ll i, ll j, std::vector<std::string> &image, std::vector<std::vector<bool>> &found, ll height, ll width)
    {
        found[i][j] = true;
        ll tempi, tempj;
        for (ll k = 0; k < 4; k++)
        {
            switch (k)
            {
                case 0:
                    tempi = i - 1;
                    tempj = j;
                    break;
                    break;
                case 1:
                    tempi = i;
                    tempj = j + 1;
                    break;
                case 2:
                    tempi = i + 1;
                    tempj = j;
                    break;
                    break;
                case 3:
                    tempi = i;
                    tempj = j - 1;
                    break;
```

```
            };

            if (tempi >= 0 && tempi < height && tempj >= 0 && tempj < width && !found[tempi][tempj] && (image[tempi][tempj] == 'L' || image[tempi][tempj] == 'C')){
                floodFill(tempi, tempj, image, found, height, width);
            }
        }
    }

};

int main(){
  std::ios::sync_with_stdio(false);
  std::cin.tie(0);
  std::cout.tie(0);
  Solution S1= Solution();
  ll i, j, k;
  ll height, width;
  ll islands;
  ll tracker = 1;
  std::cin >> height >> width;
  std::vector<std::string> image(height);
  std::vector < std::vector<bool>> found(height);
  islands = 0;
  for (i = 0; i < height; i++){
    found[i].resize(width);
    std::cin >> image[i];
  }
  for (i = 0; i < height; i++){
    for (j = 0; j < width; j++){
      if (image[i][j] == 'L' && !found[i][j]){
        S1.floodFill(i,j,image,found,height,width);
        islands++;
      }
    }
  }
  std::cout << islands << "\n";
  tracker++;
  return 0;
}

```