Flood Fill Algorithm - Grid Traversal Using Directions

```
//Header Section

#include<iostream>
#include<string>
#include<vector>

using namespace std;

//Functions

void flood ( int sr, int sc, vector<vector<int>>& maze, string& psf)
{
	if ( sr == maze.size()-1 && sc == maze[0].size()-1 )
	{
	    cout << psf << endl;
	    return;
	}

	maze[sr][sc] = 2;

	//t -> top
	if ( sr > 0 && maze[sr-1][sc] != 1 && maze[sr-1][sc] != 2 )
	{
	  psf  += "t";
	  flood ( sr-1, sc, maze, psf );
	  psf.erase(psf.length() - 1, 1);
	}

	//l -> left
	if ( sc > 0 && maze[sr][sc-1] != 1 && maze[sr][sc-1] != 2 )
	{
	  psf += "l";
	  flood ( sr, sc-1, maze, psf);
	  psf.erase(psf.length() - 1, 1);
	}

	//d -> down
	if ( sr < maze.size()-1 && maze[sr+1][sc] != 1 && maze[sr+1][sc] != 2 )
	{
	  psf += "d";
	  flood ( sr+1, sc, maze, psf );
```

```
        psf.erase(psf.length() - 1, 1);

    }

    //r -> right
    if ( sc < maze[0].size()-1 && maze[sr][sc+1] != 1 && maze[sr][sc+1] != 2 )
    {
       psf += "r";
      flood ( sr, sc+1, maze, psf );
      psf.erase(psf.length() - 1, 1);

    }

    maze[sr][sc] = 0;
}

//Main Function

int main( int argc, char** argv)
{
    vector<vector<int>> maze = {
                    {0, 1, 0, 0, 0, 0, 0, 1},
                    {0, 1, 0, 1, 1, 1, 0, 1},
                    {0, 1, 0, 1, 0, 0, 0, 1},
                    {0, 0, 0, 0, 0, 1, 1, 1},
                    {0, 1, 0, 1, 0, 0, 0, 0},
                    {0, 1, 0, 1, 1, 1, 1, 0},
                    {0, 1, 0, 1, 1, 1, 1, 0},
                    {0, 1, 0, 0, 0, 0, 0, 0}
                   };
    string psf = "";
    flood( 0 , 0 , maze , psf );
}
```