No of enclaves in C++

```
#include <iostream>
#include <vector>

using namespace std;

void dfs(vector<vector<int>>& arr, int i, int j) {
    if (i < 0 || j < 0 || i >= arr.size() || j >= arr[0].size()
|| arr[i][j] == 0) {
        return;
    }
    arr[i][j] = 0;
    dfs(arr, i + 1, j);
    dfs(arr, i - 1, j);
    dfs(arr, i, j + 1);
    dfs(arr, i, j - 1);
}

int numEnclaves(vector<vector<int>>& arr) {
    int m = arr.size();
    int n = arr[0].size();

    // Marking connected components touching the
boundaries
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            if ((i == 0 || j == 0 || i == m - 1 || j == n - 1) &&
arr[i][j] == 1) {
                dfs(arr, i, j);
            }
        }
    }

    // Counting remaining land cells
    int count = 0;
    for (int i = 0; i < m; ++i) {
        for (int j = 0; j < n; ++j) {
            if (arr[i][j] == 1) {
                ++count;
            }
        }
    }

    return count;
}

int main() {
    int m = 4, n = 4;
    vector<vector<int>> arr = {
        {0, 0, 0, 0},
        {1, 0, 1, 0},
        {0, 1, 1, 0},
        {0, 0, 0, 0}
    };

    int result = numEnclaves(arr);
    cout << result << endl;

    return 0;
}
```

Dry Run

Input Grid:

```
{
    {0, 0, 0, 0},
    {1, 0, 1, 0},
    {0, 1, 1, 0},
    {0, 0, 0, 0}
}
```

Step 1: DFS from Boundary Cells

- **Boundary cells:** We start by scanning the boundary cells (first and last rows, first and last columns). The boundary cells are:
    - Row 0: {0, 0, 0, 0}
    - Row 3: {0, 0, 0, 0}
    - Column 0: {1, 0, 0, 0}
    - Column 3: {0, 0, 0, 0}
- The boundary cells that are 1 (land) are:
    - (1, 0)

Step 2: Marking Land Cells Connected to Boundary

1. **DFS starting at (1, 0):**
    - Mark arr[1][0] as 0.
    - Explore its neighbors (down: (2, 0), left: out of bounds, right: (1, 1), up: (0, 0)).
    - No other connected land cells.

Step 3: Count Remaining Land Cells

After marking the connected land cells to the boundary, the grid looks like this:

```
{
    {0, 0, 0, 0},
    {0, 0, 1, 0},
    {0, 1, 1, 0},
    {0, 0, 0, 0}
}
```

Now, we count the remaining land cells (1) in the grid:

- (1, 2), (2, 1), and (2, 2) are the remaining land cells.

Final Answer:

The number of enclosed land cells is 3.

Output:

	3
Output:- 3	