No of enclaves in C++

```cpp
#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;
}
```

📜 Input Grid:

	0	**1**	**2**	**3**
0	0	0	0	0
1	1	0	1	0
2	0	1	1	0
3	0	0	0	0

📊 Dry Run Table – Step-by-Step

🟡 Step 1: Mark boundary-connected 1s using DFS

Check all boundary cells and run DFS from any land (`1`) on the edge:

Cell	Is Boundary?	Is Land?	DFS Run?	Action
(0,x)/(x,0)/(3,x)/(x,3)	✅ Yes	Mixed	✅ If land	DFS removes (1,0) only

✅ Only **(1,0)** is a boundary land → DFS marks it and its connected land `0`.

🔄 After DFS update, grid becomes:

	0	**1**	**2**	**3**
0	0	0	0	0
1	0	0	1	0
2	0	1	1	0
3	0	0	0	0

🟢 Step 2: Count remaining 1s (enclaves)

Cell	Value	Is Land (1)?	Count += 1?
(1,2)	1	✅	✅ (count=1)
(2,1)	1	✅	✅ (count=2)

Cell	Value	Is Land (1)?	Count += 1?
(2,2)	1	✅	✅ (count=3)

🧮 Total enclave land cells = **3**

✅ Final Output:

```
3
```

🔁 Summary Table:

Phase	Operation	Result
Boundary DFS	Remove all 1s connected to boundary	(1,0) set to 0
Enclave Counting	Count remaining 1s in the grid	3
Final Return Value	`numEnclaves()`	3

Output:-

```
3
```