Island Perimeter in C++

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

int perimeter(vector<vector<int>>& grid) {
    int p = 0;
    int rows = grid.size();
    int cols = grid[0].size();

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            if (grid[i][j] == 1) {
                p += 4;

                if (i > 0 && grid[i - 1][j] == 1) {
                    p -= 2;
                }
                if (j > 0 && grid[i][j - 1] == 1) {
                    p -= 2;
                }
            }
        }
    }
    return p;
}

int main() {
    vector<vector<int>> grid = {
        {1, 0, 0},
        {1, 1, 1},
        {0, 1, 0},
        {0, 1, 0}
    };

    int p = perimeter(grid);
    cout << p << endl;

    return 0;
}
```

Input Grid:

```
grid = {
    {1, 0, 0},
    {1, 1, 1},
    {0, 1, 0},
    {0, 1, 0}
};
```

Visualized:

```
1 0 0
1 1 1
0 1 0
0 1 0
```

🔁 Dry Run Strategy:

- Each land cell contributes +4 to perimeter.
- Each shared edge with another land cell subtracts 2.

🔍 Dry Run Table:

Cell (i,j)	grid[i][j]	+4	Top Neighbor = 1	Left Neighbor = 1	Net Contribution
(0,0)	1	4	✗	✗	4
(1,0)	1	4	✓ (0,0)	✗	2 (4-2)
(1,1)	1	4	✗	✓ (1,0)	2 (4-2)
(1,2)	1	4	✗	✓ (1,1)	2 (4-2)
(2,1)	1	4	✓ (1,1)	✗	2 (4-2)
(3,1)	1	4	✓ (2,1)	✗	2 (4-2)

✅ Total Perimeter:

= 4 + 2 + 2 + 2 + 2 + 2 = 14

✅ Output:

14