717 1-bit and 2-bit Characters (link)

Description

We have two special characters:

- The first character can be represented by one bit `0`.
- The second character can be represented by two bits (`10` or `11`).

Given a binary array `bits` that ends with `0`, return `true` if the last character must be a one-bit character.

Example 1:

```
Input: bits = [1,0,0]
Output: true
Explanation: The only way to decode it is two-bit character and one-bit character.
So the last character is one-bit character.
```

Example 2:

```
Input: bits = [1,1,1,0]
Output: false
Explanation: The only way to decode it is two-bit character and two-bit character.
So the last character is not one-bit character.
```

Constraints:

- `1 <= bits.length <= 1000`
- `bits[i]` is either `0` or `1`.

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```
#include <vector>

class Solution {
public:
    bool isOneBitCharacter(std::vector<int>& bits) {
        int i = 0;
        while (i < bits.size() - 1) {
            if (bits[i] == 0) {
                // Переходим к следующему биту (однобитный символ)
                i++;
            } else {
                // Пропускаем следующий бит (двухбитный символ)
                i += 2;
            }
        }
        // Если i равен размеру массива минус один, последний символ - однобитный
        return i == bits.size() - 1;
    }
};
```