Lab 2.9.3 Palindromes once again

Objectives

Improve the student's skills in:

- manipulating vectors;
- using the **for** loop to iterate through vectors.

Scenario

As far as we remember, you're already familiar with palindromes. Now the palindromes have returned, but this time as vectors, not just ints. The task, however, remains the same – you need to check if a declared vector contains a palindrome.

Look at the code below. There is a vector – this time you can hard-code it instead of inputting its values every time the code is launched, but don't forget to recompile your source after any amendments.

A variable named *n* has also been assigned (a bit tricky) with a value indicating the number of elements in the vector. Feel free to add and/or remove some of the vector's elements – *n* will respond flexibly to your changes.

Complete the code with a smart **for** loop, checking whether or not there is a palindrome. You don't need any new vectors to achieve the goal. Be smart. Be creative.

Check if your code works equally well when the vector stores an even and an odd number of elements.

```
#include <iostream>

using namespace std;

int main(void) {

 int vector[] = {1, 7, 3, 8, 3, 7, 1};
 bool ispalindrome = true;
 int n = sizeof(vector) / sizeof(vector[0]);

 // Insert your code here

 if(ispalindrome)
  cout << "It's a palindrome";
 else
  cout << "It isn't a palindrome";
 cout << endl;
 return 0;
```