Exercice - 5.25

Valentin Ricard

November 28, 2022

1 Informations

2 Code

2.1 exo_5_25.cpp

```cpp
//
// Created by ValentinRicard on 08.11.2022.
//

#include <algorithm>
#include <iostream>
#include <vector>

using namespace std;

ostream &operator<<(ostream &os, const vector<int> &tableau) {
    for (int value: tableau) {
        cout << value << " ";
    }
    return os;
}

bool contains(const vector<int> &vec, int value) {
    return find(vec.begin(), vec.end(), value) != vec.end();
}

bool equals(const vector<int> &vec1, const vector<int> &vec2) {
    return all_of(vec1.begin(), vec1.end(), [&vec2](int value) {
                return contains(vec2, value);
            }) &&
            all_of(vec2.begin(), vec2.end(), [&vec1](int value) {
                return contains(vec1, value);
            });
}


int main() {
    vector<int> tableau1 = {1, 3, 5, 7, 11};
    vector<int> tableau2 = {1, 3, 5, 7, 12};
    vector<int> tableau3 = {11, 1, 1, 3, 5, 7, 11};

    cout << boolalpha << "t1: " << tableau1
         << ", t2: " << tableau2
         << ", t3: " << tableau3
         << endl;

    cout << "t1=t2: " << equals(tableau1, tableau2)
```

```
         << ", t1=t3: " << equals(tableau1, tableau3);
}
```

2.2 exo_5_25.cpp

```cpp
//
// Created by ValentinRicard on 08.11.2022.
//

#include <algorithm>
#include <iostream>
#include <vector>

using namespace std;

ostream &operator<<(ostream &os, const vector<int> &tableau) {
    for (int value: tableau) {
        cout << value << " ";
    }
    return os;
}

bool contains(const vector<int> &vec, int value) {
    return find(vec.begin(), vec.end(), value) != vec.end();
}

bool equals(const vector<int> &vec1, const vector<int> &vec2) {
    return all_of(vec1.begin(), vec1.end(), [&vec2](int value) {
               return contains(vec2, value);
           }) &&
           all_of(vec2.begin(), vec2.end(), [&vec1](int value) {
               return contains(vec1, value);
           });
}


int main() {
    vector<int> tableau1 = {1, 3, 5, 7, 11};
    vector<int> tableau2 = {1, 3, 5, 7, 12};
    vector<int> tableau3 = {11, 1, 1, 3, 5, 7, 11};

    cout << boolalpha << "t1: " << tableau1
         << ", t2: " << tableau2
         << ", t3: " << tableau3
         << endl;

    cout << "t1=t2: " << equals(tableau1, tableau2)
         << ", t1=t3: " << equals(tableau1, tableau3);
}
```