Домашно СД, Група 1

Задача 1. Напишете функция `longestSeq`, която работи върху итератори на произволна последователност, приема предикат и връща итератор към началото на най-дългата подредица от последователни елементи, за които предикатът е изпълнен. При няколко подредици с еднаква максимална дължина върнете началото на първата. Ако няма нито един елемент, удовлетворяващ предиката, върнете `last` (крайния итератор). Демонстрирайте функцията с `std::vector` и `std::list`.

Пример (обикновен `std::vector` и предикат).

```
std::vector<int> v{1, 2, 4, 6, 7, 8, 10, 12, 14, 1, 3, 5};
auto it = longestSeq(v.begin(), v.end(),
                     [](int x){ return x % 2 == 0; });

// Очакване: it == v.begin() + 5 (стойност 8).
// Най-дългата последователност е: 8, 10, 12, 14.
```

Задача 2. Имплементирайте клас `TinySet` — множество от *цели числа*, което съхранява **до 256 различни елемента**. Елементите се пазят **сортирани във възходящ ред** и **без повторения**.

Класът трябва да има **поне** следните методи:

- `bool add(int x);` — добавя `x`, ако не присъства. Връща `true` при успешно добавяне и `false` при вече съществуващ елемент или при изчерпан капацитет (256).
- `bool removeIfExists(int x);` — премахва `x`, ако съществува. Връща `true` при успешно премахване и `false` в противен случай.
- `bool contains(int x) const;` — проверява дали `x` принадлежи на множеството.

Реализирайте итератори за обхождане на множеството, така че да са съвместими със стандартните алгоритми и диапазонни `for`-цикли:

- `TinySet::iterator` и `TinySet::const_iterator` с методи `begin()`, `end()`, `cbegin()`, `cend()`.
- `TinySet::reverse_iterator` (`const_reverse_iterator`) с методи `rbegin()`, `rend()` (`crbegin()`, `crend()`).