

Задачи:

1. Реализирайте двойно свързан цикличен списък, като поддържате следните методи:
 - isEmpty()
 - size()
 - front()
 - back()
 - assign(size_t n, T element) – инициализира списъка с n на брой копия на element
 - pushFront(T element) – добавя елемент в началото
 - pushBack()
 - popFront()
 - popBack()
 - insert(size_t position, T element)
 - erase(size_t position)
 - clear()
 - unique() – remove duplicate values
 - reverse()
 - removeIf(bool (*pred)(T element)) – изтрива всички елементи, които удовлетворяват предиката
 - map(T (*func)(T element))

Реализирайте и итератор за нашия свързан списък ☺

2. Skip List (basic вариант)

```
template <class T>
struct node
{
    T data;
    node<T> *next, *skip;
}
```

Използваме реализацията на свързан списък с указател next. **Елементите на Skip List са под дадена наредба.** Някои елементи могат да съдържат в указателя skip адреса на друг елемент, намиращ се по-напред в редицата. В нашия Skip List искаме всеки \sqrt{n} –ти елемент да има указател към следващия \sqrt{n} -ти елемент. Реализирайте следните методи на класа Skip List:

- insertSorted() – добавя елементи стандартно, ползвайки само next указателя
- speedUp() – построява бързите връзки в списъка, след като в него са вмъкнати определен брой елементи
- find(T element) – връща позицията на element или -1 ако няма такъв
- copy constructor – да работи и за бързите връзки!!!
- operator= – да работи и за бързите връзки!!!