

Занятие 13. Списки

Обратите внимание, что для тестирования функций, представленных в этом задании, будет удобно написать несколько вспомогательных функций, для таких операций как создание и сравнение списков.

В заданиях можно использовать списки с выделенной головой или без неё. Тип возвращаемого значения для функций в первом и втором задании определите согласно типу списка.

Задание 1

Напишите функцию

```
reverse(shared_ptr<Node> head);
```

которая переворачивает односвязный список (т.е. элемент, который хранился в первом узле списка, после работы функции окажется последним, второй – предпоследним и т.д.). На вход функции подаётся указатель на голову списка.

Функция не должна выделять новую динамическую память, т.е. использовать `make_shared` или `new`.

Задание 2

Напишите функцию

```
removePart(shared_ptr<Node> list1, shared_ptr<Node> list2);
```

которая проверяет, что односвязный список `list2` входит в `list1`, и удаляет ту часть списка `list1`, которая совпадает со списком `list2`. На вход подаются указатели на головы списков `list1` и `list2`.

Если список `list1` не содержит список `list2`, функция не должна ничего делать.

Задание 3

Напишите шаблонную функцию

```
template<class InputIt>  
shared_ptr<Node2> insertSort(InputIt first, InputIt last);
```

которая создаёт двухсвязный список с заданными элементами, отсортированными в порядке возрастания. Данные для сортировки задаются с помощью пары итераторов: `first` и `last` (элемент `last` не входит в диапазон), и, таким образом, могут находиться в любой структуре данных `stl`. Функция должна создать двунаправленный список и поместить в него элементы массива в порядке возрастания, используя сортировку вставками. Функция возвращает указатель на созданный список.

Например, после выполнения такого кода:

```
std::vector<int> data = {5, 3, 20, 8};
shared_ptr<Node> list = insertSort(data.cbegin(), data.cend());
```

список `list` должен иметь вид $3 \leftrightarrow 5 \leftrightarrow 8 \leftrightarrow 20$.

Оценка задания

Задание 1	33%
Задание 2	33%
Задание 3	36%

Баллы за каждую задачу включают:

- правильность решения задания 60 %;
- оформление кода согласно <http://prog.tversu.ru/pr3/codeStyle.pdf> 10 %;
- программа компилируется без предупреждений компилятора на 4-ом уровне 10%;
- модульные тесты 20%.