

1. Для односвязного и двусвязного списка написать функцию удаления из списка всех элементов (вместе с освобождением памяти), которые удовлетворяют предикату и возвращает указатель на новый первый элемент списка

```
ListNode* RemoveAll(ListNode* first, DataCompareFunc predicate);
```

В качестве примера предиката использовать функции:

- a. Значение Salary \geq 500.
- b. Name начинается с 'M'
- c. Любой ваш вариант условия на поле Salary и/или Name

2. Для односвязного списка написать функцию обращения (смены порядка на обратный), возвращающую новое значение первого узла

```
ListNode* Revertl(ListNode* first);
```

3. Для односвязного списка написать функцию сортировки пузырьком, в качестве критерия сравнения двух элементов передавать указатель на функцию-предикат

```
typedef int (*TwoDataCompareFunc) (const Data*, const Data*);
```

, которая возвращает -1, если первый аргумент меньше второго, 1 – если первый аргумент больше второго и 0 – если они равны.

Сама функция сортировки будет иметь вид:

```
void BubbleSort(ListNode* first, TwoDataCompareFunc predicate);
```

При написании функции использовать вспомогательную функцию:

```
void Swap(ListNode* node1, ListNode* node2);
```

Функция не меняет сами узлы местами, а только указатели на данные, которые в них хранятся.

4. Написать пример, демонстрирующий сортировку 1) по полю Name, 2) по полю Salary.

5. Написать пример, демонстрирующий работу с функциями RemoveAll, Revert.