

Лабораторная 1

1. Создать проект из заголовочного и исполнимого файлов в архиве Lab1.zip.
2. Подготовить проект к тестированию работы с динамической памятью. Для этого в исходном файле source.cpp поставить точки останова в строках с номерами 10, 19, 46, 53, 60, 62.
3. Запустить проект в режиме Debug, на первой точке останова настроить режим управления снимками памяти, сделать первый снимок и поместить в окно контрольных значений переменные l1, p, l2, t.
4. Перемещаясь от одной точки останова к другой через команду «продолжить», получать в каждой точке останова снимки памяти и следить за изменением состояний переменных в окне просмотра контрольных значений.
5. При повторном запуске отладчика выполнить пошаговую трассировку одного из циклов создания списка и удаления списка (для упрощения можно уменьшить количество элементов в списке).
6. Оформить алгоритм удаления списка в виде функции. Функция должна принимать указатель на начало удаляемого списка по ссылке! Проверить отсутствие утечки памяти и корректность значений указателей после выполнения функции.

Остальные задания выполняются уже без контроля утечек памяти, поэтому можно изменить описание структуры elem_list, убрав из нее поле buf.

7. Раскомментировать фрагмент программы, начиная со строки 30 до строки 44. Убрать точки останова и выполнить всю программу целиком.
8. Оформить алгоритм печати списка в виде функции. Обратите внимание, что указатель на начало списка передается в функцию по значению, поэтому можно обойтись без вспомогательной переменной p.
9. Изменить в примере создание списка таким образом, чтобы можно было вводить длину списка и элементы списка с консоли.
10. Написать функции:
 - a. определения длины списка. Для пустого списка длина равна 0.
 - b. посчитать количество элементов списка с заданным значением.
 - c. определить позицию первого элемента с заданным значением (вернуть указатель на этот элемент). Если такого элемента нет вернуть nullptr.
 - d. вставить новый элемент в список за первым элементом с заданным значением. Если вставить невозможно, функция должна выбрасывать исключение.

- e. удалить первый элемент с заданным значением из списка.
- f. удалить все элементы с заданным значением.

11. Написать функцию создания списка, заданной длины из элементов массива.

Используя эту функцию написать тесты для функции определения длины списка, функции определения количества элементов списка с заданным значением.

12. Следующие функции работы со списками реализовать рекурсивно:

- a. найти сумму значений элементов списка, сделать тесты.
- b. посчитать количество элементов списка с заданным значением, сделать тесты.
- c. определить максимальное значение в списке, сделать тесты.
- d. напечатать элементы списка в обратном порядке.