**Лабораторная работа №1. Обработка списков**

Необходимо реализовать предложенный вариант задания. Номер задания студента определяется порядковым номером студента в журнале старосты.

Требования к отчету:

1. Титульный лист
2. Задание по варианту
3. Описание реализации
4. Описание возникших затруднений
5. Описание альтернативных способов решения

Требования к программе:

1. Ввод элементов списка должен быть доступен с клавиатуры и путем автоматической генерации
2. Программа должна корректно обрабатывать некорректный ввод.
3. Исходный код должен быть откомментирован
4. Необходимо реализовать правильную декомпозицию программы на методы.
5. Необходимо реализовать предложенный алгоритм без использования стандартных функций и с использованием (необходимо реализовать два отдельных метода).

Варианты заданий:

1.

Из списка удалить самую длинную цепочку четных элементов.

Пример: из списка A[8]: 4 1 4 2 1 2 4 6 должен получиться список A[5]: 4 1 4 2 1

(самая длинная цепочка четных чисел включает элементы с 6 по 8: 2 4 6).

2.

Из списка удалить цепочки из четных элементов, состоящие менее чем из трех элементов.

Пример: из списка A[8]: 4 3 4 2 1 2 4 6 должен получиться список A[5]: 3 1 2 4 6.

3.

Из списка удалить цепочки из нечетных элементов, состоящие менее чем из трех элементов.

Пример: из списка A[8]: 3 3 4 5 2 3 7 9 должен получиться список A[5]: 4 2 3 7 9.

4.

Из списка A удалить те элементы, которые встречаются и в списке A и в списке B по крайней мере по 2 раза.

Пример: список A[8]: 3 3 4 5 2 3 5 9

список B[7]: 1 2 3 4 5 2 5

По 2 раза в обоих списках встречается только элемент, равный 5.

список A после удаления примет вид: A[6]: 3 3 4 2 3 9.

5.

Из списка из каждой цепочки четных элементов удалить самый маленький элемент.

Пример: из списка A[9]: 3 6 4 5 2 3 4 6 4 должен

получиться список A[6]: 3 6 5 3 6 4.

6.

Из списка A удалить те цепочки четных элементов, в которых есть хотя бы один элемент из списка B.

Пример: список A[9]: 3 2 4 5 2 3 2 6 5

список B[6]: 1 3 4 7 8 9

список A после удаления примет вид:

A[7]: 3 5 2 3 2 6 5.

7.

Из списка A удалить те цепочки нечетных элементов, в которых нет ни одного элемента из списка B.

Пример: список A[10]: 3 2 7 5 2 1 2 6 3 9

список B[5]: 1 2 5 4 8

список A после удаления примет вид:

A[7]: 2 7 5 2 1 2 6.

8.

Из списка A удалить те цепочки нечетных элементов, в которых нет ни одного элемента из списка B.

Пример: список A[10]: 3 2 7 5 2 1 2 6 3 9

список B[5]: 1 2 5 4 8

список A после удаления примет вид:

A[7]: 2 7 5 2 1 2 6.

9.

Между списками A и B обменять их самые длинные цепочки из одинаковых элементов.

Пример: список A[10]: 3 2 2 5 2 1 1 1 3 9

список B[8]: 1 2 5 5 4 8 3 3

В списке A самая длинная цепочка:

1 1 1 (элементы с 7 по 9).

В списке B самая длинная цепочка:

5 5 (элементы с 3 по 4).

список A после перестановки в него цепочки из списка B: A[9]: 3 2 2 5 2 5 5 3 9.

список B после перестановки в него цепочки из списка A: B[9]: 1 2 1 1 1 4 8 3 3.

10.

Между списками A и B обменять их самые длинные цепочки из четных элементов.

Пример: список A[10]: 3 2 4 6 2 1 1 1 8 9

список B[7]: 1 0 5 5 4 3 3

В списке A самая длинная цепочка:

2 4 6 2 (элементы со 2 по 5).

В списке B самая длинная цепочка:

0 (элемент 2).

список A после перестановки в него цепочки из списка B:

A[7]: 3 0 1 1 1 8 9.

список B после перестановки в него цепочки из списка A:

B[10]: 1 2 4 6 2 5 5 4 3 3.

11.

Из списка удалить четные элементы, стоящие после максимального.

Пример: из списка A[5]: 2 7 4 6 5 должен получиться

список A[3]: 2 7 5.

12.

Из списка удалить четные элементы, имеющие значение больше среднего арифметического всех элементов списка.

Пример: из списка A[5]: 8 7 2 6 5 должен получиться

список A[3]: 7 2 5 (среднее арифметическое всех элементов =(8+7+2+6+5)/5=5.6).

13.

Из списка удалить элементы, имеющие значение меньше среднего арифметического четных элементов списка.

Пример: из списка A[5]: 8 7 2 6 5 должен получиться

список A[3]: 8 7 6 (среднее арифметическое четных элементов =(8+2+6)/3=5.33).

14.

Из списка удалить элементы, стоящие после максимального и имеющие значение меньше среднего арифметического всех элементов списка.

Пример: из списка A[5]: 8 6 9 4 5 должен получиться

список A[3]: 8 6 9 (среднее арифметическое четных элементов

=(8+6+9+4+5)/5=6.4).

15.

Из списка удалить четные элементы, стоящие между максимальным и минимальным элементами.

Пример: из списка A[7]: 1 8 8 4 7 0 5 должен получиться список A[5]: 1 8 7 0 5.

16.

Из списка удалить элементы, кратные трем, стоящие между максимальным и минимальным элементами.

Пример: из списка A[7]: 1 9 3 4 9 0 0 должен получиться список A[5]: 1 9 4 0 0.

17.

Из списка удалить элементы, имеющие четный индекс и стоящие между максимальным и минимальным элементами.

Пример: из списка A[7]: 9 3 4 9 1 0 0 должен получиться список A[5]: 9 4 1 0 0.

18.

Из списка удалить элементы, встречающиеся в списке более одного раза.

Пример: из списка A[7]: 9 3 4 9 1 0 0 должен получиться список A[3]: 3 4 1.

19.

Из списка удалить элементы, встречающиеся в списке только один раз.

Пример: из списка A[7]: 9 1 4 9 1 9 0 должен получиться список A[5]: 9 1 9 1 9.

20.

Из списка удалить нечетные элементы, встречающиеся в списке только один раз.

Пример: из списка A[7]: 4 1 4 3 1 9 0 должен получиться список A[5]: 4 1 4 1 0.

21. Реализовать сортировку слиянием.

22. Реализовать быструю сортировку.

23. Реализовать пузырьковую сортировку.

24. Реализовать сортировку вставками.

25. Реализовать сотртировку выбором.