Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования «Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

**Отчет по лабораторной работе №1**

Дисциплина: «Разработка профессиональных приложений»

Обработка списков

Вариант № 5

Выполнил:

студент группы ИВТАПбд-21

Гаязов Т. Р.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 5 вариант задания, а именно: из списка удалить минимальные элементы в цепочке четных чисел.

**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.

**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).

**Описание реализации**

**Общее:**

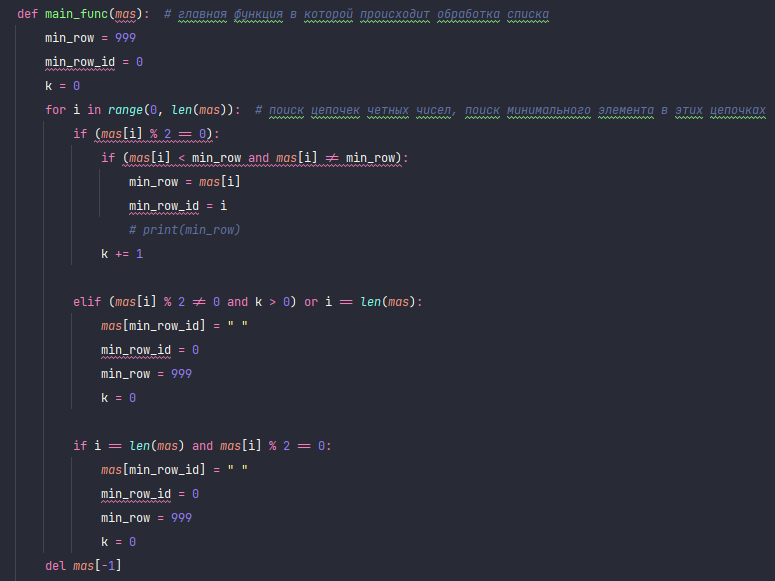
Были созданы функции manual\_filling() и manual\_filling\_wo\_func(mas), внутри которых происходит ввод списка с клавиатуры с использованием стандартных функций и без использования стандартных функций соответственно.

Для рандомного заполнения импортируем библиотеку random, вызываем ее метод randint(0, 100) который заполнит наш список рандомной длины(от 5 до 10) целочисленными значениями от 0 до 99.

Для ручного заполнения создаем цикл for, который выполняется до тех пор, пока длина нашего списка не станет равна длине списка.

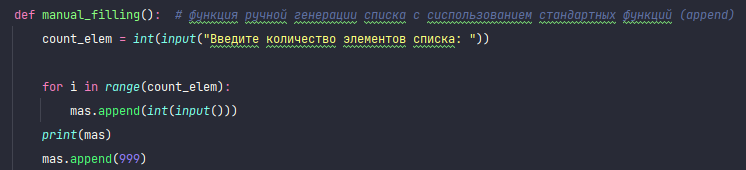
В нашей программе нам необходимо найти минимальный элемент в каждой цепочке четных элементов списка. Затем найденные элементы удалить из списка и вывести получившийся список.

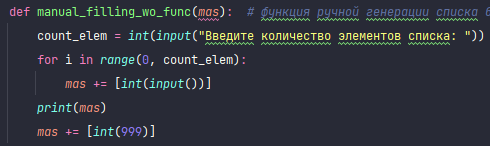
В функции main\_func(mas) происходит поиск цепочек четных чисел и поиск минимального элемента в этих цепочках



Удаление нужных элементов происходит в цикле while благодаря методу del, который удаляет элемент по его индексу.

Функция ручной генерации списка с использованием стандартных методов(append)



Функция ручной генерации списка без использования стандартных методов

**Описание возникших затруднений**

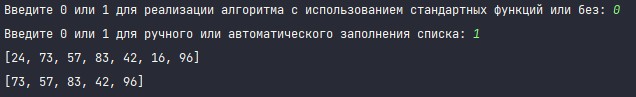
При написании кода трудностей не возникло.

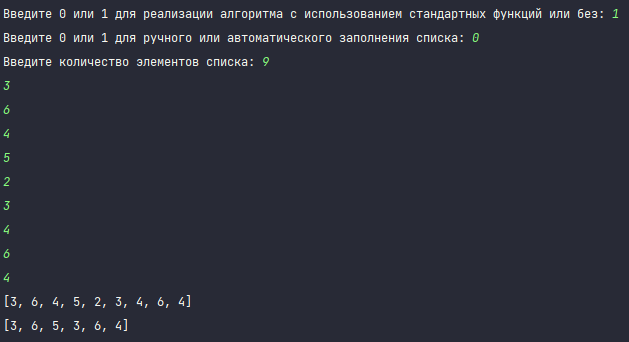
**Описание альтернативных способов решения**

Ключевое слово del при удалении можно заменить методами списка – pop() и remove() отличие лишь в том, что remove() удаляет элемент по первому его вхождению, pop() возвращает удаленный элемент по его индексу, а del удаляет элемент по индексу. Так как нам не нужно возвращать удаленный элемент было выбрано ключевое слово del.

Выбран метод append() вместо extend() так как нам необходимо на каждой итерации цикла while добавлять 1 элемент, а не расширять список добавляя в него новый объект.

**Результаты работы программы**

**1рис. Рандомное заполнение списка со стандартными методами.**

****

**2. рис. Заполнение с клавиатуры списка без стандартных методов.**

**Код программы**

import random  
import sys  
  
while True:  
 mas = []  
  
 check\_method = *int*(*input*("Введите 0 или 1 для реализации алгоритма с использованием стандартных функций или без: "))  
 check = *int*(*input*("Введите 0 или 1 для ручного или автоматического заполнения списка: "))  
  
  
 def manual\_filling(): *# функция ручной генерации списка с сиспользованием стандартных функций (append)* count\_elem = *int*(*input*("Введите количество элементов списка: "))  
  
 for i in *range*(count\_elem):  
 mas.append(*int*(*input*()))  
 *print*(mas)  
 mas.append(999)  
  
  
 def manual\_filling\_wo\_func(*mas*): *# функция ручной генерации списка без использования стандартных функций* count\_elem = *int*(*input*("Введите количество элементов списка: "))  
 for i in *range*(0, count\_elem):  
 *mas* += [*int*(*input*())]  
 *print*(*mas*)  
 *mas* += [*int*(999)]  
  
  
 min\_row\_id = 0  
  
  
 def main\_func(*mas*): *# главная функция в которой происходит обработка списка* min\_row = 999  
 min\_row\_id = 0  
 k = 0  
 for i in *range*(0, *len*(*mas*)): *# поиск цепочек четных чисел, поиск минимального элемента в этих цепочках* if (*mas*[i] % 2 == 0):  
 if (*mas*[i] < min\_row and *mas*[i] != min\_row):  
 min\_row = *mas*[i]  
 min\_row\_id = i  
 *# print(min\_row)* k += 1  
  
 elif (*mas*[i] % 2 != 0 and k > 0) or i == *len*(*mas*):  
 *mas*[min\_row\_id] = " "  
 min\_row\_id = 0  
 min\_row = 999  
 k = 0  
  
 if i == *len*(*mas*) and *mas*[i] % 2 == 0:  
 *mas*[min\_row\_id] = " "  
 min\_row\_id = 0  
 min\_row = 999  
 k = 0  
 del *mas*[-1]  
 ii = 0  
 while " " in *mas*: *# удаление элементов списка, являющиеся минимум в своих цепочках* if *mas*[ii] == " ":  
 del *mas*[ii]  
 ii += 1  
  
 *print*(*mas*)  
 *print*(" ")  
  
  
 def with\_func(*mas*): *# определение метода генерации списка* if check == 0:  
 if check\_method == 0:  
 manual\_filling()  
 main\_func(*mas*)  
 elif check\_method == 1:  
 manual\_filling\_wo\_func(*mas*)  
 main\_func(*mas*)  
  
  
 elif check == 1:  
 if check\_method == 0:  
 mas = [random.randint(0, 100) for i in *range*(random.randint(5, 10))]  
 *print*(*mas*)  
 *mas*.append(999)  
 elif check\_method == 1:  
 mas = [random.randint(0, 100) for i in *range*(random.randint(5, 10))]  
 *print*(*mas*)  
 *mas* += [*int*(999)]  
 main\_func(*mas*)  
 else:  
 *print*("Ошибка ввода")  
 sys.exit()  
  
  
 *# определение метода (с использованием стандартных функций или без)* if check\_method == 0:  
 with\_func(mas)  
 elif check\_method == 1:  
 with\_func(mas)  
 else:  
 *print*("Введите только 0 или 1")