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 № 1

Выполнил:

студент группы ИВТАПбд-21

Аронов В. В.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту**

Необходимо реализовать 1 вариант задания, а именно: 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.

**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:**

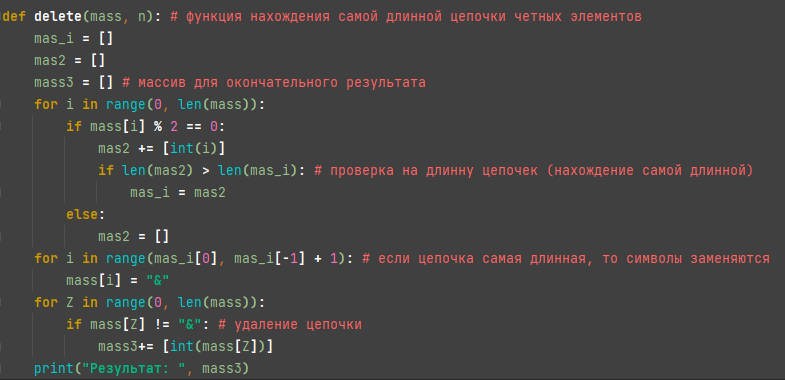
Были созданы 2 файла, в которых реализована работа программы без использования стандартных функций и с использованием стандартных функций.

Для рандомного заполнения импортируем библиотеку random, вызываем ее метод randint(0, 100) который заполнит наш список рандомной длины(от 3 до 8) 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.

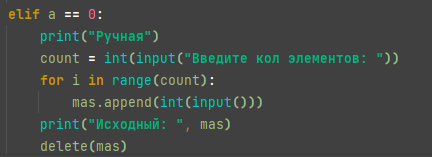
В нашей программе нам необходимо найти самую длинную цепочку четных элементов. Затем найденные элементы удалить из списка и вывести получившийся список.

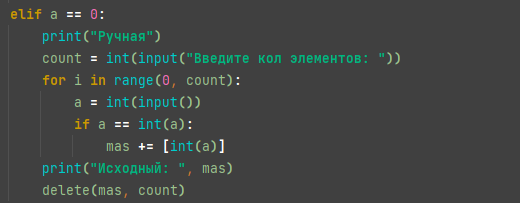
В функции delete(mass, n) происходит поиск цепочек четных чисел



Удаление нужных элементов происходит в цикле благодаря замене цепочек символом “&” и занесением в новый словарь списка без этих символов.

Функция ручного ввода списка с использованием стандартных методов(append)



Функция ручного ввода списка без использования стандартных методов 

**Описание возникших затруднений**

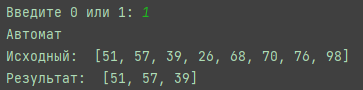
При написании кода трудностей не возникло.

**Описание альтернативных способов решения**

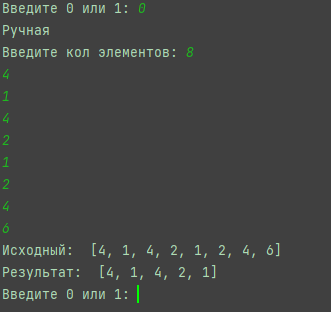
Ключевое слово del при удалении можно заменить методами списка – pop() и remove() отличие лишь в том, что remove() удаляет элемент по первому его вхождению, pop() возвращает удаленный элемент по его индексу, а del удаляет элемент по индексу. Так как нам не нужно возвращать удаленный элемент было выбрано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

****

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

****

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

**Код программы**

**Без использования стандартных функций**

**import** random  
**import** sys  
  
  
**def delete(**mass, n**):** # функция нахождения самой длинной цепочки четных элементов  
 mas\_i **= []** mas2 **= []** mass3 **= []** # массив для окончательного результата  
 **for** i **in** range**(**0, len**(**mass**)):  
 if** mass**[**i**] %** 2 **==** 0**:** mas2 **+= [**int**(**i**)]  
 if** len**(**mas2**) >** len**(**mas\_i**):** # проверка на длинну цепочек (нахождение самой длинной)  
 mas\_i **=** mas2  
 **else:** mas2 **= []  
 for** i **in** range**(**mas\_i**[**0**]**, mas\_i**[-**1**] +** 1**):** # если цепочка самая длинная, то символы заменяются  
 mass**[**i**] = "&"  
 for** Z **in** range**(**0, len**(**mass**)):  
 if** mass**[**Z**] != "&":** # удаление цепочки  
 mass3**+= [**int**(**mass**[**Z**])]** print**("Результат: "**, mass3**)  
  
  
while True:** a **=** int**(**input**("Введите 0 или 1: "))** mas **= []  
 if** a **==** 1**:** count **=** 0  
 print**("Автомат")** mas **= [**random.randint**(**0, 100**) for** i **in** range**(**random.randint**(**3, 8**))]** print**("Исходный: "**, mas**)** delete**(**mas, count**)  
 elif** a **==** 0**:** print**("Ручная")** count **=** int**(**input**("Введите кол элементов: "))  
 for** i **in** range**(**0, count**):** a **=** int**(**input**())  
 if** a **==** int**(**a**):** mas **+= [**int**(**a**)]** print**("Исходный: "**, mas**)** delete**(**mas, count**)  
 else:** print**("error")** sys.exit**()**

**С использованием стандартных функций**

**import** random  
**import** sys  
  
  
**def delete(**mass**):** # функция нахождения самой длинной цепочки четных элементов  
 mas\_i **= []** mas2 **= []** mass3 **= []** # массив для окончательного результата  
 **for** i **in** range**(**0, len**(**mass**)):  
 if** mass**[**i**] %** 2 **==** 0**:** mas2.append**(**i**)  
 if** len**(**mas2**) >** len**(**mas\_i**):** # проверка на длинну цепочек (нахождение самой длинной)  
 mas\_i **=** mas2  
 **else:** mas2 **= []  
 for** i **in** range**(**mas\_i**[**0**]**, mas\_i**[-**1**] +** 1**):** # если цепочка самая длинная, то символы заменяются  
 mass**[**i**] = "&"  
 for** Z **in** range**(**0, len**(**mass**)):  
 if** mass**[**Z**] != "&":** # удаление цепочки  
 mass3.append**(**mass**[**Z**])** print**("Результат: "**, mass3**)  
  
  
while True:** a **=** int**(**input**("Введите 0 или 1: "))** mas **= []** mas3 **= []  
 if** a **==** 1**:** print**("Автомат")** mas **= [**random.randint**(**0, 100**) for** i **in** range**(**random.randint**(**3, 8**))]** print**("Исходный: "**, mas**)** delete**(**mas**)  
 elif** a **==** 0**:** print**("Ручная")** count **=** int**(**input**("Введите кол элементов: "))  
 for** i **in** range**(**count**):** mas.append**(**int**(**input**()))** print**("Исходный: "**, mas**)** delete**(**mas**)  
 else:** print**("error")** sys.exit**()**