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 №7

Выполнила:

студентка группы ИВТАПбд-22

Ренжина А.С.

Проверил:

преподаватель кафедры

«Вычислительная техника»

Исхаков И.И.

Ульяновск, 2023

**Задание по варианту:** 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.

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:**

Первым делом пишем условие, если файл называется main, тогда он запускает функцию с идентичным названием.

В функции main, прописываем способы ввода элементов списка, а именно - вручную или путем автоматической генерации. Для этого создаем метод, в котором реализуем условия, при вводе цифры, которому соответствует одно из условий, и выполняет. При ручном вводе, реализуем ввод с клавиатуры, а путем автоматической реализации, заполнение массива с помощью библиотеки random.

Далее в методе main, передаем метод action - это метод, содержащий в себе основную часть работы кода. Там при помощи условий, проверяются те условия, которые нам нужны и удаляются элементы, если они совпали. Для этого создаем методы check и del\_el, первая отвечает за проверку того, что в цепочке нечетных элементов нет элементов из второго списка, то нам возвращает True, или же False. Второй метод удаляет нужные нам элементы.

На выходе получаем конечный результат. Так как в методе main есть цикл while True, и ввод 0 - закончить программу, то программа будет работать до того, пока мы не введем 0 или же закончим сами.

**Тестовые примеры:**

**Пример 1.**

1 - ввод вручную

2 - автоматический ввод

0 - закончить программу

2

Список А: [8, 4, 7, 4, 1, 5, 8, 6, 3, 6, 4, 5, 5]

Список B: [6, 7, 1, 2, 0, 9, 10]

Конечный результат: [8, 4, 7, 4, 1, 5, 8, 6, 6, 4]

**Пример 2.**

1 - ввод вручную

2 - автоматический ввод

0 - закончить программу

1

5 9 1 5 4 3 6

5 6 2 1

Список А: [5, 9, 1, 5, 4, 3, 6]

Список B: [5, 6, 2, 1]

Конечный результат: [5, 9, 1, 5, 4, 6]

**Пример 3.**

1 - ввод вручную

2 - автоматический ввод

0 - закончить программу

2

Список А: [7, 5, 2, 0, 10, 5, 6, 3, 2, 8, 10, 2, 5, 6, 9, 9, 0]

Список B: [1, 2, 4, 6, 6, 9, 1, 1, 0]

Конечный результат: [2, 0, 10, 6, 2, 8, 10, 2, 6, 9, 9, 0]

1 - ввод вручную

2 - автоматический ввод

0 - закончить программу

0

Process finished with exit code 1

**Исходный код:**

import random

def check(a, b, start, end): # проверяет, что элемент из списка А, не встречается в списке В,

flag = False # и если встречается, то возвращает false

for j in a[start:end]:

if j not in b and j % 2 != 0:

flag = True

else:

return False

return flag

def del\_el(a, start, end): #функция удаления из списка А элементов, подходящие условию

m = end - 1

while m >= start:

if a[m] % 2 != 0:

a.pop(m)

m -= 1

return a

def action(a, b): #основная функция, содержащая методы проверки и удаления

start = -1

end = 0

i = 0

while i < len(a):

if a[i] % 2 != 0 and start == -1:

start = i

else:

i += 1

if i + 1 < len(a) and a[i + 1] % 2 == 0 and start != -1:

end = i + 1

if check(a, b, start, end):

a = del\_el(a, start, end)

i = 0

else:

i += 1

start = -1

end = 0

if i + 1 == len(a):

if a[i] % 2 != 0:

if start == -1:

start = i

end = i + 1

if check(a, b, start, end):

del\_el(a, start, end)

i += 1

return a

def input\_arr(itype): #метод выбора ввода

a = []

b = []

if itype == 1:

a = list(map(int, input().split()))

b = list(map(int, input().split()))

if itype == 2:

n = random.randint(10, 20)

for i in range(n):

a.append(random.randint(0, 10))

n = random.randint(5, 10)

for i in range(n):

b.append(random.randint(0, 10))

if itype == 0:

exit(1)

return a, b

def main():

while True:

print("1 - ввод вручную\n"

"2 - автоматический ввод\n"

"0 - закончить программу")

itype = int(input())

a, b = input\_arr(itype)

print(f"Список А: {a}")

print(f"Список B: {b}")

a = action(a,b)

print(f"Конечный результат: {a}")

if \_\_name\_\_ == "\_\_main\_\_":

main()

**Описание возникших затруднений:**

При выполнении лабораторной работы, затруднений не возникло.

**Описание альтернативных способ решения:**

Альтернативных способ решения при выполнении лабораторной работы не возникло.

**Вывод о проделанной работе:**

Во время выполнения лабораторной работы, познакомились со списками и как их обрабатывать, а также работу с несколькими методами.