**Федеральное агентство связи  
Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования   
«Московский технический университет связи и информатики»**

Кафедра «Теории вероятностей и прикладной математики»

**Лабораторная работа №4**

По дисциплине «Структуры и алгоритмы обработки данных»   
по теме «Реализация стека\дека»  
Вариант №6

Выполнила: Гердт Таисия

Группа: БСТ1955

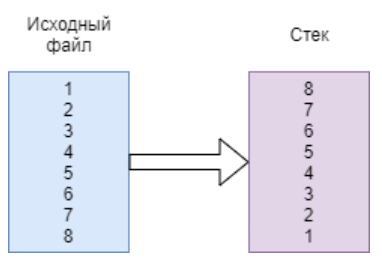
Руководитель: Кутейников Иван Алексеевич

## Задание

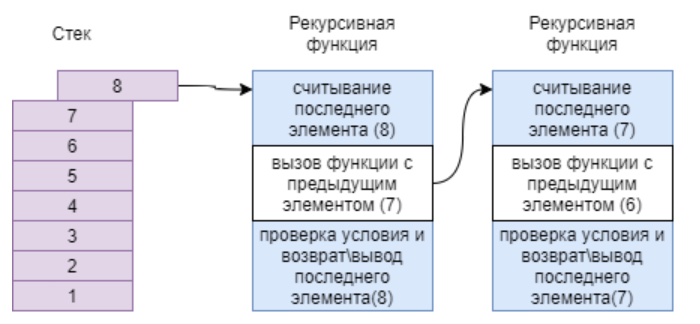
Используя технологию модульного программирования разработать программу обработки данных, содержащихся в заранее подготовленном файле, в соответствии с индивидуальным заданием. Применить динамическую структуру указанного в задании вида: стек, очередь или дек. Программа должна включать модуль, содержащий набор всех необходимых средств (типов, подпрограмм и т.д.) для решения поставленной задачи.

Вариант 6. Дан файл из вещественных чисел. Используя стек за один просмотр файла напечатать сначала все числа, меньшие a, затем все числа из интервала [a,b], и, наконец, все остальные числа, сохраняя исходный порядок в каждой группе.

Стек (от англ. stack — стопка) — структура данных, представляющая из себя упорядоченный набор элементов, в которой добавление новых элементов и удаление существующих производится с одного конца, называемого вершиной стека (по принципу LIFO, от англ. last in — first out, «последним пришёл — первым вышел»).



Так как задание требуется выполнить за один просмотр, данные в стеке будут храниться в обратном порядке. Для того, чтобы выводить данные в прямом порядке, воспользуемся рекурсивной функцией:



Для реализации стека создадим новый класс, хранящий массив данных, имеющий методы добавления и получения элементов, метод проверки на пустоту. В рекурсивную функцию будем отправлять стек и лямбду-функцию, задающую проверку выполнения условия.

Код программы:

# новый класс для стека

class Stack():

#пустой массив для хранения значений

array = []

#метод стека – проверка на пустоту массива внутри класса

def empty(self):

if len(self.array) == 0:

return True

else:

return False

#метод стека – добавление нового элемента к массиву

def put(self, item):

self.array.append(item)

#метод стека – вывод последнего элемента и его удаление в массиве

def get(self):

return self.array.pop()

#метод – вывод всех элементов массива из stack, удовлетворяющих условию func (с последнего элемента до первого, при помощи рекурсии)

def pop(stack, func):

if not stack.empty():

num = stack.get()

pop(stack, func)

if func(num):

print(num)

else:

stack.put(num)

mystack = Stack()

a = float(input("Enter a:\n"))

b = float(input("Enter b:\n"))

#чтение чисел из файла без переноса строки и пробелов и их запись в стек

with open("file.txt", mode="r", encoding="utf-8-sig") as f:

for lines in f:

for l in lines.rstrip('\n').split(' '):

mystack.put(float(l))

print(f"\nLess than {a}:")

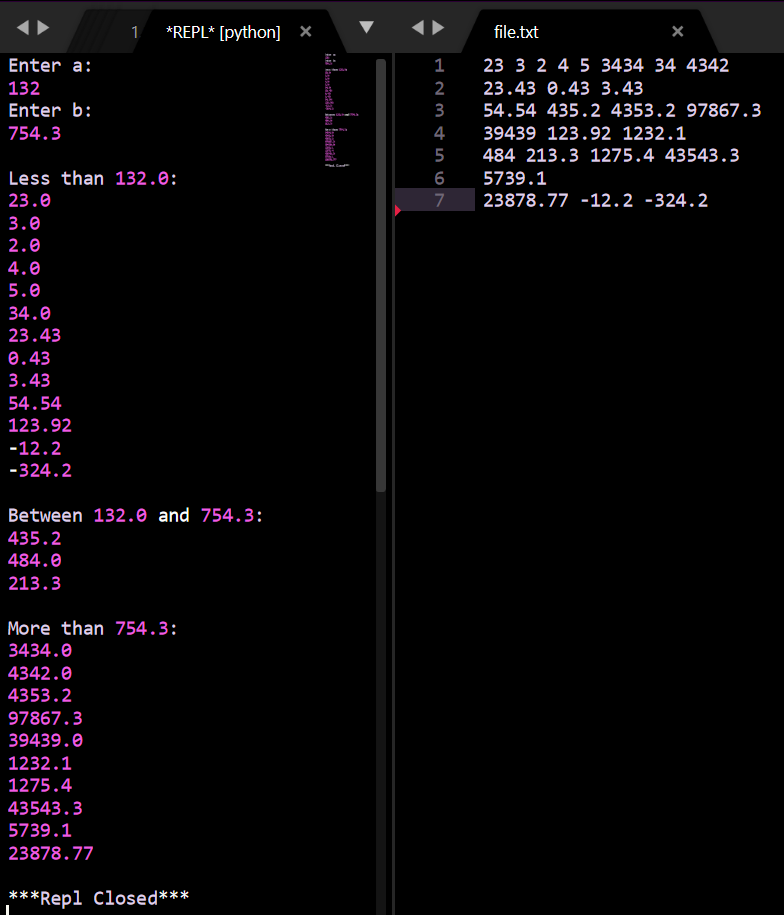
pop(mystack, lambda x: x < a)

print(f"\nBetween {a} and {b}:")

pop(mystack, lambda x: x <= b)

print(f"\nMore than {b}:")

pop(mystack, lambda x: x > b)

Результаты выполнения программы:

Вывод: Реализация стека осуществляется с помощью рекурсии; с помощью стека за один просмотр возможно осуществить вывод элементов, удовлетворяющих заданным условиям.