

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ

Языки программирования

Лабораторная работа №6 Строки

Руководитель

подпись, дата

А.В. Проскурин
инициалы, фамилия

Обучающийся БПИ23-02, 23151451
номер группы, зачетной книжки

подпись, дата

С.А. Черкашин
инициалы, фамилия

Красноярск 2024 г.

ЦЕЛЬ РАБОТЫ

Получение практических навыков разработки и отладки программ, работающих с текстовыми файлами.

ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучите теоретические сведения к данной лабораторной работе.
2. Ознакомиться с постановкой задачи (вариант задания вычисляется как $\text{ваш_номер_в_списке_группы} \% \text{ количество_вариантов}$).
3. Составить программу, решающую поставленную задачу. ***Все функции разместить в отдельном файле (модуле)***, который будет подключаться в основном проекте.
4. Отладить программу на подготовленных наборах тестовых данных.
5. Подготовить отчет по лабораторной работе. Отчет должен включать в себя:
 - a. постановку задачи;
 - b. текст программы с комментариями;
 - c. наборы тестовых исходных данных с соответствующими результатами работы программы;
 - d. краткие ответы на контрольные вопросы
6. Защитить лабораторную работу перед преподавателем

Создать текстовый файл, в котором будут храниться элементы исходных списков. В отдельном модуле написать три функции:

Вариант 2. Дан список вещественных значений $A[N]$. Определить количество и индексы тех элементов списка, которые меньше элементов, стоящих в соседних позициях справа и слева, но больше минимального элемента списка.

ХОД РАБОТЫ

Func.py

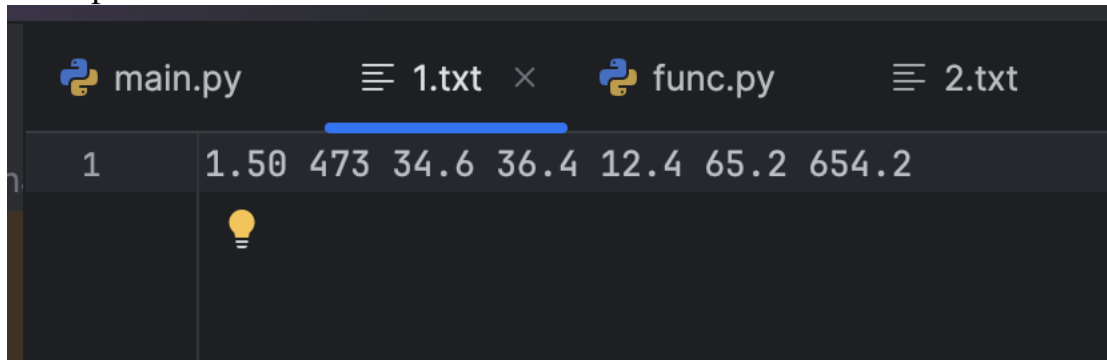
```
main.py 1.txt func.py 2.txt
1
2
3 def readFile(fileName): 1 usage
4     file = open(fileName, "r")
5     contentFile = file.read()
6     content = contentFile.split(" ")
7     toFloat = [float(i) for i in content]
8     file.close()
9     return toFloat
10
11
12 def mainFunc(mas): 1 usage
13     minNum = min(mas)
14     count = 0
15     index = []
16     for i in range(1, len(mas) - 1):
17         if mas[i - 1] > mas[i] > minNum and mas[i] < mas[i + 1]:
18             count += 1
19             index.append(i+1)
20     return count, index
21
22
23 def writeToFile(fileName, count, index): 1 usage
24     file = open(fileName, "w")
25     file.write(f"Кол-во чисел: {count}\n")
26     file.write("Индексы: ")
27     for i in index:
28         file.write(f"{i} ")
29
```

Main.py

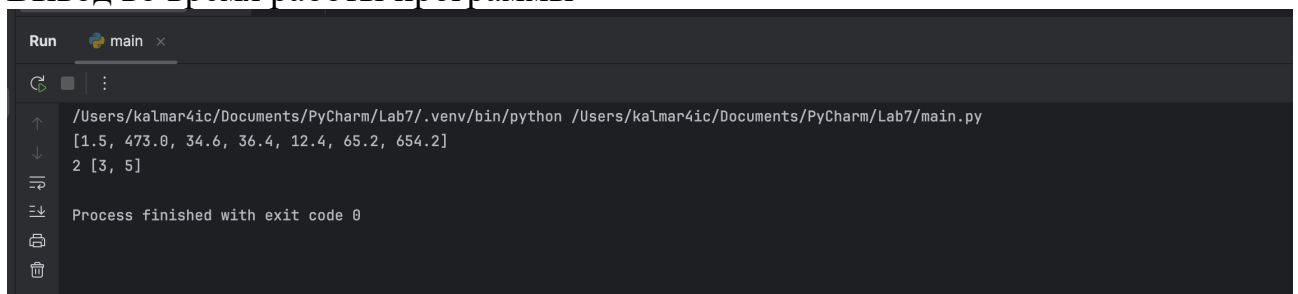
```
main.py 1.txt func.py 2.txt
1 import func
2
3 text = func.readFile("1.txt")
4 print(text)
5 count, index = func.mainFunc(text)
6 print(count, index)
7 func.writeToFile(fileName: "2.txt", count, index)
8
```

Демонстрация работы программы

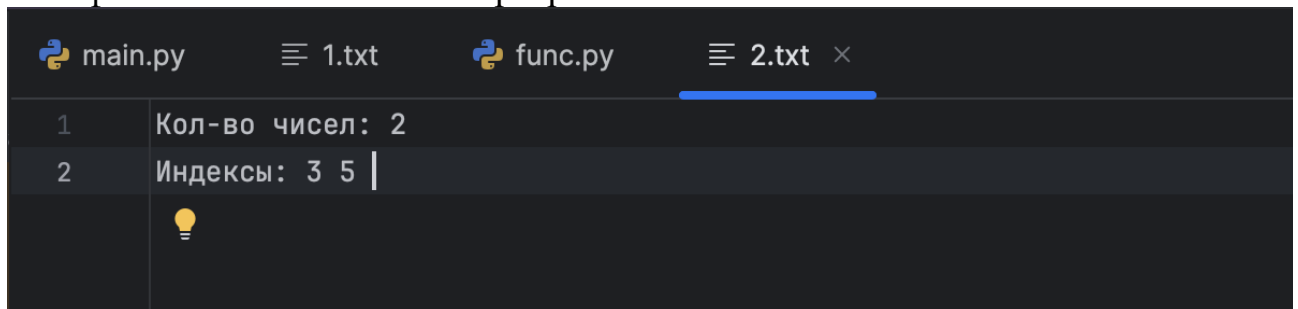
1.txt файл



Вывод во время работы программы



2.txt файл после выполнения программы



ОТВЕТЫ НА КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Какие типы файлов существуют? В чём их особенность?

Существуют два основных типа файлов: текстовые и бинарные. Текстовые файлы хранят только строковую информацию (тип `str` в языке Python) и открываются в любом текстовом редакторе. Бинарные файлы хранят информацию так же, как она хранится в оперативной памяти компьютера, но нужно знать размер каждого сохранённого данного и порядок их записи в файле.

2. Какая последовательность действий при работе с файлом?

Открыть файл, считать с него данные, закрыть файл.

3. Какие параметры необходимо указать при открытии файла?

Имя файла, режим доступа, кодировка.

4. В чём разница между относительным и абсолютным путями к файлу?

Для относительного указывается имя файла относительно папки, в которой находится программа. Для абсолютного указывается полный путь к файлу, начиная с корневого каталога.

5. Какие существуют режимы доступа для открытия файла?

`r` – для чтения, `w` – для записи, `a` – для добавления в конец, `t` – текстовый режим, `b` – бинарный режим, `+` – для чтения и записи

6. Какие методы существуют для чтения данных и в чём разница между ними?

Метод `read()` считывает весь файл в одну строку, ему можно указать значение количества символов которые считать в одну строку. Метод `readlines()` считывает весь файл построчно в список строк. Метод `readline()` считывает текущую строку, при следующем вызове будет считана следующая.

7. В чём преимущество использования менеджера контекста?

Преимущество этого способа в том, что файл всегда корректно закрывается после выполнения блока, либо если при выполнении была получена ошибка.

8. В чём отличия работы с текстовым и бинарным файлами?

Бинарные файлы, в отличие от текстовых, хранят информацию в виде набора байт. Для работы с ними в Python необходим встроенный модуль `pickle`. При открытии бинарного файла на чтение или запись нужно применять режим «`b`» в дополнение к режиму записи («`w`») или чтения («`r`»).

9. Как добавить и считать данные из бинарного файла?

`dump(obj, file)` – записывает объект `obj` в бинарный файл `file`. `load(file)` – считывает данные из бинарного файла в объект.

ВЫВОДЫ

Были получены практические навыки работы с текстовыми файлами.