МИНЕСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧЕРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«Брестский государственный технический университет»

Кафедра «Интеллектуальные информационные технологии»

Лабораторная работа №2

По дисциплине «Современные системы программирования»

За 7 семестр

Тема: «Работа с файловой системой»

Выполнила:  
студентка 4 курса  
группы АС-56  
Карпенко М.В.

Проверил:

Войцехович Г.Ю.

Брест 2022

*Цель работы:* Приобрести практические навыки обработки параметров командной строки, закрепить базовые знания языка программирования Python при решении практических задач.

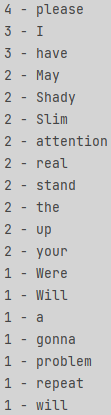
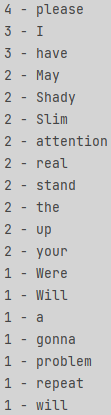
ВАРИАНТ 4

*Задание 1.* Напишите программу, выполняющую чтение текстовых данных из файла и их последующую обработку: 4) Напишите программу, которая выводит слова, располагая их в порядке убывания частоты их появления. Перед каждым словом должно быть число его появлений.

*Код программы:*

def first\_task():  
 words = []  
 check = ""  
 file = open("Text.txt")  
 for line in file:  
 line = re.sub(r'[^\w\s]', '', line)  
 for symbol in line:  
 if symbol.isalpha():  
 check += symbol  
 else:  
 words.append(check)  
 check = ""  
 words.sort()  
  
 lsWord = {}  
 for key in words:  
 if key in lsWord:  
 value = lsWord[key]  
 lsWord[key] = value + 1  
 else:  
 lsWord[key] = 1  
  
 sorted\_keys = sorted(lsWord, key=lambda x: int(lsWord[x]), reverse=True)  
  
 for key in sorted\_keys:  
 s = str("{0} - {1}").format(lsWord[key], key)  
 print(s)

*Результат работы программы:*

* *

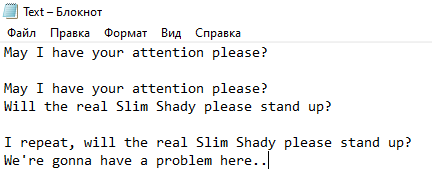
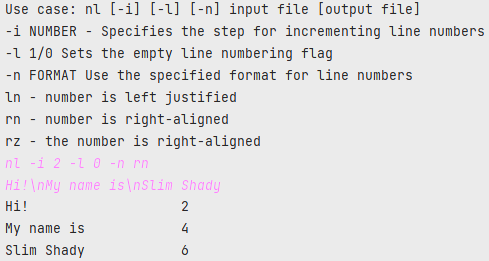
*Задание 2.* Утилита nl выводит переданный файл в стандартный вывод или в другой файл, выполняя нумерацию его строк. Если файл не задан или задан как –, читает стандартный ввод.

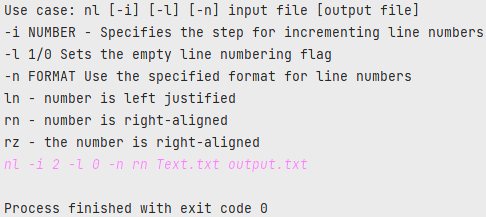
Формат использования: nl [-i] [-l] [-n] входной\_файл [выходной\_файл]

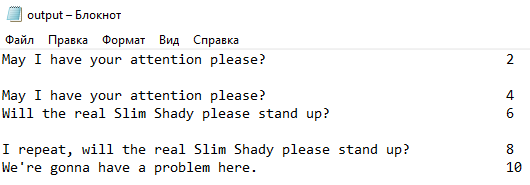
*Код программы:*

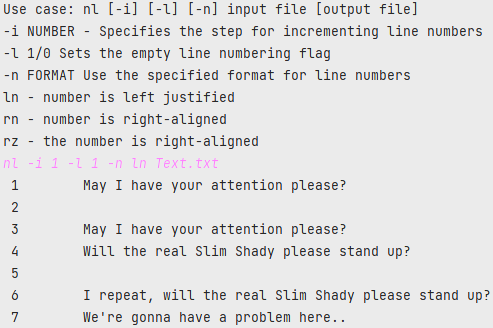
def main():  
 print("Use case: nl [-i] [-l] [-n] input file [output file]")  
 print("-i NUMBER - Specifies the step for incrementing line numbers\n"  
 "-l 1/0 Sets the empty line numbering flag\n"  
 "-n FORMAT Use the specified format for line numbers\n"  
 "ln - number is left justified\n"  
 "rn - number is right-aligned\n"  
 "rz - the number is right-aligned")  
 line = input()  
  
 index\_i = line.find("-i")  
 index\_i += 2  
 if line[index\_i] == " ":  
 index\_i += 1  
  
 index\_l = line.find("-l")  
 index\_l += 2  
 if line[index\_l] == " ":  
 index\_l += 1  
  
 index\_n = line.find("-n")  
 index\_n += 2  
 if line[index\_n] == " ":  
 index\_n += 1  
  
 step = line[index\_i]  
 flag = line[index\_l]  
 line\_format = line[index\_n:index\_n + 2]  
  
 files = line[index\_n + 3:]  
 count = files.count(".")  
 if count == 2:  
 input\_file = files[:files.find(" ")]  
 output\_file = files[files.find(" ") + 1:]  
 elif count == 1:  
 new\_count = files.find(".")  
 k = files.find(" ")  
 if new\_count < k:  
 input\_file = files[:files.find(" ")]  
 output\_file = "None"  
 else:  
 input\_file = "None"  
 output\_file = files[files.find(" ") + 1:]  
 else:  
 input\_file = "None"  
 output\_file = "None"  
  
 if input\_file != "None":  
 file\_in = open(input\_file, "r")  
 text = []  
 for i in file\_in:  
 text.append(i)  
 final = {}  
 keys = []  
 a = 0  
 length = 0  
 key = "key"  
 for j in range(len(text)):  
 if text[j] == "\n" and flag == "0":  
 key += step  
 final[key] = text[j]  
 keys.append(key)  
 else:  
 a += int(step)  
 final[a] = text[j]  
 keys.append(a)  
 if len(text[j]) > length:  
 length = len(text[j])  
 else:  
 text\_line = input()  
 final = {}  
 keys = []  
 a = 0  
 length = 0  
 while True:  
 f = text\_line.find("\\n")  
 stroke = text\_line[:f]  
 if len(stroke) + 1 == len(text\_line):  
 a += int(step)  
 final[a] = text\_line  
 keys.append(a)  
 break  
 a += int(step)  
 final[a] = stroke  
 text\_line = text\_line[f + 2:]  
 keys.append(a)  
 if len(stroke) > length:  
 length = len(stroke)  
 length += 10  
 if output\_file == "None":  
 for i in range(len(keys)):  
 if line\_format == "ln":  
 if isinstance(keys[i], str):  
 print(" " + final.get(keys[i]), end="")  
 else:  
 print("{:< 10}".format(keys[i]) + final.get(keys[i]), end="")  
 else:  
 if isinstance(keys[i], str):  
 print(final.get(keys[i])[:-1] + " ")  
 else:  
 different = length - len(final.get(keys[i])[:-1])  
 print(final.get(keys[i])[:-1] + " " \* different + "{:< 10}".format(keys[i]))  
 else:  
 file\_out = open(output\_file, 'w')  
 for i in range(len(keys)):  
 if line\_format == "ln":  
 if isinstance(keys[i], str):  
 file\_out.write(" " + final.get(keys[i]))  
 else:  
 file\_out.write("{:< 10}".format(keys[i]) + final.get(keys[i]))  
 else:  
 if isinstance(keys[i], str):  
 file\_out.write(final.get(keys[i])[:-1] + " " + "\n")  
 else:  
 different = length - len(final.get(keys[i])[:-1])  
 file\_out.write(final.get(keys[i])[:-1] + " " \* different + "{:< 10}".format(keys[i]) + "\n")  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

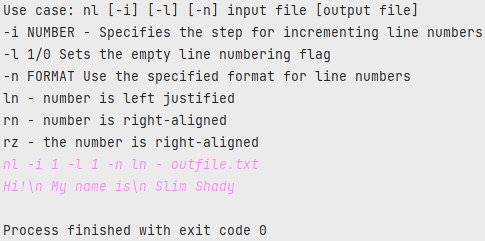
*Результат работы программы:*

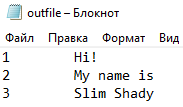
 











*Вывод:* Приобрела практические навыки обработки параметров командной строки, закрепила базовые знания языка программирования Python при решении практических задач.