Федеральное государственное автономное

образовательное учреждение высшего образования

«Омский государственный технический университет»

Кафедра "Информатика и вычислительная техника"

**ОТЧЕТ**

**ЛАБОРАТОРНАЯ РАБОТА №2**

|  |  |
| --- | --- |
|  | Выполнил: |
|  | студент группы ПИН-222 Сорокина С.Д. |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (дата, подпись) |
|  | Проверил: |
|  | ассистент Елисеенко Ю.А. |
|  |  |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (дата, подпись) |

Омск 2024

**ОПИСАНИЕ ЗАДАЧИ**

Задача 3.4 (Обычная перестановка)

Даны две строки a и b вывести строку х максимальной длины, состоящую из, букв, таких, что существует произвольная перестановка х, являющаяся подстрокой произвольной перестановки строки a и одновременно являющаяся подстрокой произвольной перестановки строки b.

Задача 4.4 (Аладушки)

Приготовить идеальную стопку оладий - это хитрое дело, потому что, как вы ни старайтесь, все оладьи в стопке имеют разные диаметры. Тем не менее, для аккуратности вы можете упорядочить стопку по размеру так, чтобы каждая оладья была меньше всех оладий, находящихся под ней. Размер оладьи определяется ее диаметром.

Сортировка стопки производится серией «переворотов» оладий. Переворот состоит в том, что вы помещаете лопатку между двумя оладьями в стопке и переворачиваете (меняете порядок на обратный) все оладьи на лопатке (реверсируете подстопку). Переворот задается позицией оладьи, находящейся внизу подстопки, которую нужно перевернуть по отношению ко всей стопке. Позиция нижней оладьи 1, тогда как для стопки из n оладий позиция верхней оладьи = n.

Стопка определяется заданием диаметра каждой оладьи в стопке в порядке следования. Например, рассмотрим три стопки оладий, причем в левой стопке оладья 8 является самой верхней:

8 7 2

4 6 5

6 4 8

7 8 4

5 5 6

2 2 7

От левой стопки к средней можно перейти путем flip(3). От средней стоики к правой можно перейти путем команды flip(l).

**БЛОК-СХЕМЫ**

Блок-схема решения задачи 3.4 представлена на рисунке 1. Функция normalPermutation принимает на вход список words содержащий в себе все слова, считанные из тестового файла.

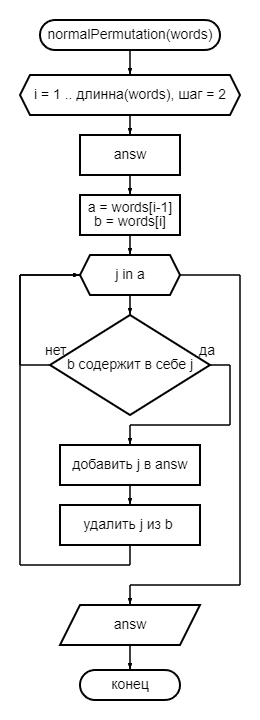


Рисунок 1 – Блох-схема функции normalPermutation

Блок-схема решения задачи 4.4 представлена на рисунке 2.

Рисунок 2 – Блох-схема функций orderDeeck и makeDeck

**НАБОРЫ ТЕСТОВЫХ ДАННЫХ**

Содержимое файлов для тестирования программ, решающих задачи 3.4 и 4.4 представлены на рисунках 3 и 4 соответственно.

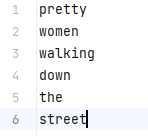


Рисунок 3.1 – Файл для тестирования программы, решающий задачу 3.4

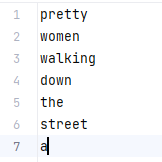


Рисунок 3.2 – Файл для тестирования программы, решающий задачу 3.4

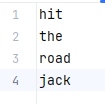


Рисунок 3.3 – Файл для тестирования программы, решающий задачу 3.4

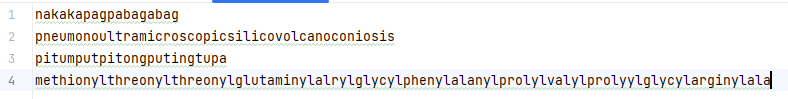


Рисунок 3.4 – Файл для тестирования программы, решающий задачу 3.4

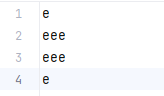


Рисунок 3.5 – Файл для тестирования программы, решающий задачу 3.4

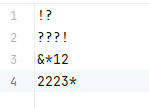


Рисунок 3.6 – Файл для тестирования программы, решающий задачу 3.4

Рисунок 4.2 – Файл для тестирования программы, решающий задачу 2.4

Рисунок 4.3 – Файл для тестирования программы, решающий задачу 2.4

Рисунок 4.4 – Файл для тестирования программы, решающий задачу 2.4

Рисунок 4.5 – Файл для тестирования программы, решающий задачу 2.4

**КОД НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON**

Код, решающий задачи 3.4 и 4.4 представлен в листингах 1 и 2.

def readFile(text):

f = open(text, 'r', encoding="UTF-8")

words = []

while True:

line = f.readline().strip()

if not line:

break

words.append(line)

if len(words)%2==0:

return words

else:

raise ValueError("Количество строк поступивших на вход нечетное")

def normalPermutation(words):

for i in range(1, len(words), 2):

answ = []

a = list(words[i-1])

b = list(words[i])

for j in a:

if j in b:

answ.append(j)

b.pop(b.index(j))

print("".join(sorted(answ)))

from os import listdir, getcwd

print(getcwd())

files = [i for i in listdir(getcwd()) if ".txt" in i]

for i, file in enumerate(files, start=1):

try:

print(f"Тест {i}")

normalPermutation(readFile(file))

except Exception as e:

print(e)

Листинг 1 – Код программы, решающий задачу 3.4

Листинг 2 – Код программы, решающий задачу 4.4

**РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММ**

Скриншоты с результатами работы программ представлены в рисунках 5 и 6.

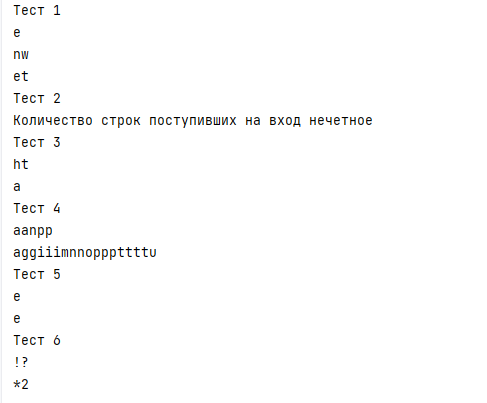


Рисунок 5 – Результат работы программы, решающей задачу 3.4

Рисунок 6 – Результат работы программы, решающей задачу