ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ

за курсом «Основи Об’єктно-орієнтованого програмування»

студентки групи ПС-20-1

Тесленко Анастасія Петрівна

кафедра комп’ютерних технологій, ДНУ

2021/2022

1. **Постановка задачі :**

Мені потрібно було зробити програму вхідні дані якої - це параметри командного рядка. Послідовність слів тексту. Якщо користувач не задав параметри командного рядка,то програма входить в діалог з користувачем і питає про ім’я файлу, з якого треба прочитати вхідні дані. Перша функція другого модуля упорядковує на дану їй через список параметрів послідовність за неспаданням суми цифр дробної частини числа(або суми рядка матриці, або реверсованогослова тексту).

Критерій –це функція, передана такожякпараметрдляфункціїупорядкування.

Друга функція другого модуля має для заданої послідовності(або матриці) обчислити і повернути чотири результати : максимум, мінімум, та їх номери(індекси) у заданій послідовності(матриці).

І в кінці програма мусить видати результати у текстовий файл, ім’я якого вона отримує з командного рядка.

1. **Опис розв’язку:**

На початку роботи після вивчення умови задачі я почала думати як має в мене виглядати програма: це точно головне меню, далі потрібно буде зробити функції для реверсу слів, для сортування, для пошуку мінімального та максимального елементів. Але спочатку потрібно зробити зчитування нашої послідовності та першим елементом буде в нас назва файлу, в який ми потім запишемо результати, бо так було сказано в умові задачі.

Отже, почала з того що потрібно вивести скільки елементів написав користувач та вивести ці елементи з перевіркою чи є вони там або чи достатньо їх там. Далі написати як називається файл, в який записуємо, а якщо не буде послідовності, то спитати де її можна знайти та записуємо вміст файлу у вектор. Далі вже потрібно почати робити функції, почала я з myReverse Ідея була така, що потрібно поділити слово на 2, наприклад, в нас є схсх після ділення буде: сх сх у temp записуємо с, далі ось С міняємо місцями з останнім, отже: хх сс, далі буде теж саме, але з наступною буквою: хс хс. Далі пострібно було зробити вже саме сортування, отже, в функції сортування ми реверсимо нашу послідовність та потрібно зробити лічильники для слів та символів у кожному слові та перевіряємо символи, тобто, який перший за алфавітом. І той, що більше переміщаємо у темп і так далі. А коли відсортували, то виконувати знову функцію реверсу, щоб слова в нас стали назад, які були в умові та вивести їх. Це було моє рішення першого завдання. Далі потрібно було зробити мін та макс значення, це я зробила в новій функції getMax та getMin тут я використовувала пари, щоб брати їх та порівнювати. І потрібно задати максимальне значення та його індекс та порівнювати це значення з кожним елементом в послідовності, по якій ми проходимо і якщо елемент більший, то пишемо що він максимальний та його індекс, так само потім зробити з мінімальним, але порівнювати якщо менший елемент. Потім вже закінчення роботи, але я зрозуміла що потрібно записувати у файл, тому зробила ще функцію write в який буде відкриватись наш файл та записувати у нього наші відповіді. Далі я все записала у заголовки та почнемо вивід у головному файлі. Робимо функцію реверсу для нашого масиву слів та пишемо, що сортуємо, функція виконується та роблю так щоб були коли між словами, а в кінці крапка. Все це я ще вирішила зробити в кольорі, тому ще додаю колір. Потім виводимо мін та макс значення та в кінці записую все до файлу і якщо все виходить, то потрібно вивести, що все записано у файл і так само якщо щось не вийшло та не записано у файл.

**3. Вихідний текст програми розв’язку задачі (основні фрагменти з коментарями)**

Ось фрагмент коду коментарями де відбувається сортування:

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

//перший модуль

string myReverse(string str)//робимо функцію реверсу

{

for (int i = 0; i < str.size() / 2; i++)//довжину нашого слова ділемо на 2

{

char temp = str[i];//записуємо до чарівського temp

str[i] = str[str.size() - 1 - i];//та змінюємо місцями перший та останній елемент

str[str.size() - 1 - i] = temp;

}

return str;

}

vector <string> sortReversed(vector<string> arr)//функція сортування вже відреверсованих елементів

{

for (int i = 0; i < arr.size(); i++) //беремо нашу послідовність

{

arr[i] = myReverse(arr[i]); //робимо реверс цих елементів

}

for (int i = 0; i < arr.size(); i++)//i це лічильник для того щоб перебирати слова

{

for (int j = 0; j < arr.size() - i - 1; j++) // j це лічильник для перебору символів у кожному і-тому слові

{

if (arr[j] > arr[j + 1])

{

string temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

for (int i = 0; i < arr.size(); i++)

{

arr[i] = myReverse(arr[i]);

}

return arr;

}

Опис коду другого модуля, де відбувається пошук мінімального та максимального значень та їх індексів:

#include <iostream>

#include <string>

#include <fstream>

#include <vector>

using namespace std;

//modul2

pair<int, int> getMax(vector<string> arr)//беремо пару 2 елементів

{

int Max = 0;//нехай максимальне = 0

int indexMax = 0;//індекс максимального також

for (int i = 0; i < arr.size(); i++)//далі беремо послідовність

{

if (arr[i].size() > Max) {//та кожний елемент перевіряємо чи більше максимального

Max = arr[i].size();//якщо так, то записуємо у максимальне

indexMax = i;//та його індекс

}

}

return pair<int, int>(Max, indexMax);

}

pair<int, int> getMin(vector<string> arr)//далі беремо пари для мінімального вже

{

int Min = 1000000;//нехай це мінімум

int indexMin = 0;//а це його індекс

for (int i = 0; i < arr.size(); i++)//далі беремо послідовність

{

if (arr[i].size() < Min) {//та кожний елемент перевіряємо чи менше мінімального

Min = arr[i].size();//якщо так, то записуємо у мінімальне

indexMin = i;//та його індекс

}

}

return pair<int, int>(Min, indexMin);

}

bool write(string outputPath, vector<string> arr, pair<int, int> Max, pair<int, int> Min)//функція запису у файл

{

ofstream file(outputPath);//нехай дано файл

string line;

if (file.is\_open())//якщо його відкрито

{

file << "Result:";//пишемо у його початок наших результатів

for (int i = 0; i < arr.size(); i++)//беремо послідовність

{

file << arr[i] << " ";//записуємо її

}

file << endl

<< "Max element: " << getMax(arr).first << " | Index of max: " << getMax(arr).second; //та апис макс+індекс

file << endl

**4. Опис інтерфейсу програми (керівництво користувача)**

Програма входить в діалог ще в початку програми, якщо не буде написані слова.

if (argc < 2) {

cout << " no output parameter "; //якщо менше 2 слів, то виводимо що нічого не дано

return 0;

}

Далі в нас програма буде показувати назву файлу, в який потрібно записати все та далі буде вже записано самі відповіді на питання.

Якщо у файл в кінці програми не вийшло все записати, то буде записано: "results were not written in file" А якщо все записалось, то буде: "results were written in file".

Також у роботі було зроблено відповіді ріними кольорами

**5. Опис тестових прикладів :**

**Изображение выглядит как текст

Автоматически созданное описание**

У цьому прикладі ми бачимо, що виводить нашу послідовність та назву файлу, в який все потрібно занести, далі відсортовану послідовність( яку до цього було реверсовано та так відсортовано) та бачимо, що вивели мін та макс – це за кількістю символів в слові – це схсх та d. Та написано, що вдалося помістити все це у файл

Ось ситуація, коли нічого не написано.

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как текст

Автоматически созданное описание

А це приклад, коли по 1 символу було записано.

Изображение выглядит как текст

Автоматически созданное описание

Це коли написано тільки назва у який файл записувати, то програма питає в якому файлі написана наша послідовність.

**6. Аналіз помилок (опис усунення зауважень)**

Я хотіла спочатку зробити цілу функцію для зчитування того, що у списку аргументів, але відмовилась від цієї ідеї, бо 1. У завданні цього не було. Та 2. Це спростило мені розуміння самої програми та полегшало код сам та в кінцевому варіанті вийшло менше строк коду.

Також була проблема, що не рахувалось максимальне значення, бо в самому початку я задала що цей максимум буде дорівнювати -1, але я потім задала,. Що максимум буде дорівнювати 0 та вийшло, що програма правильно рахує ці значення.