МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 7

з навчальної дисципліни

“Базові методології та технології програмування”

ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБЛЕННЯ МАСИВІВ ДАНИХ ТА СИМВОЛЬНОЇ ІНФОРМАЦІЇ ЗА СТАНДАРТОМ UNICODE

ВИКОНАВ

студент академічної групи \_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_ Вероніка СИТЕНКОВА

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ганна ДРЄЄВА

Кропивницький – 2024

**Тема:** Програмна реалізація оброблення масивів даних та символьної інформації за стандартом unicode

Завдання до лабораторної роботи:

1. Створити персональний обліковий запис GitHub.

2. Реалізувати програмне забезпечення розв’язування задачі 7.1.

3. Реалізувати програмне забезпечення розв’язування задачі 7.2.

4. Створити Git-репозиторій для спільної роботи над проєктом з контролем версій.

**Варіант 8**

**Задача 7.1**

Користувач вводить речення (українською або англійською мовою), яке закінчується “//”, “-“ або “;”. Вивести кількість символів “й” у введеному реченні; якщо зазначений символ відсутній, вивести речення без голосних літер.

Вхідні дані: sentence[]

Вихідні дані: better[], symbol

**Алгоритм:**

Початок

1. Записати речення, яке вводить користувач

2. Порахувати кількість символу “й”

3. Якщо кількість символу > 0, то вивести їх кількість.

Інакше: вивести речення без голосних.

Кінець

**Лістинг 7.1**

#include <stdio.h>

#include <stdlib.h>

int main(){

printf("%c%c%c%c%c%c%c %c%c%c%c%c%c%c%c\n", -126, -94, -91, -92, 105, -30, -20, -32, -91, -25, -91, -83, -83, -17, 58);

char sentence[100], better[100];

int count = 0;

int symbol = 0;

fgets(sentence, sizeof(sentence), stdin);

for(int i = 0; i < sizeof(sentence); i++){

if (sentence[i] == 47 && sentence[i+1] == 47 || sentence[i] == 45 || sentence[i] == 59)

break;

else {

if (sentence[i] == -87) symbol = symbol + 1;

better[i] = sentence[i];

count = count + 1;

}

}

if (symbol == 0){

printf("%c%c%c%c%c%c%c %c%c%c %c%c%c%c%c%c%c%c%c", -112, -91, -25, -91, -83, -83, -17, -95, -91, -89, -93, -82, -85, -82, -31, -83, -88, -27, 58);

for(int i = 0; i < count; i++){

if (better[i] != 65 && better[i] != 69 && better[i] != 73 && better[i] != 79 && better[i] != 85 && better[i] != 89

&& better[i] != 97 && better[i] != 101 && better[i] != 105 && better[i] != 111 && better[i] != 117 && better[i] != 121

&& better[i] != -128 && better[i] != -123 && better[i] != -120 && better[i] != -114 && better[i] != -109 && better[i] != -98 && better[i] != -97

&& better[i] != -96 && better[i] != -91 && better[i] != -88 && better[i] != -82 && better[i] != -29 && better[i] != -18 && better[i] != -17

&& better[i] != -14 && better[i] != -13 && better[i] != -12 && better[i] != -11){

printf("%c", better[i]);

}

}

}

else printf("%c%c%c%c%c%c%c%c%c %c%c%c %c %c%c%c%c%c%c%c%c %d ", -118, 105, -85, -20, -86, 105, -31, -30, -20, 39, -87, 39, -29, -32, -91, -25, -91, -83, -83, 105, 58, symbol);

getch();

return 0;

}

**Задача 7.2**

Вхід: масив 14 чисел.

Вихід: кількість та суму додатніх елементів введеного масива.

**Алгоритм**

Початок

1. Ввод масиву чисел.

2. Якщо число позитивне додати 1 до кількості і його до суми.

3. Вивести кількість позитивних чисел та їх суму.

Кінець

**Лістинг 7.2**

#include <stdio.h>

#include <stdlib.h>

int main()

{

system("chcp 1251 & cls");

int numbers[14];

int positive = 0;

int suma = 0;

printf("Введіть 14 чисел: \n");

for(int i = 0; i < 14; i++)

scanf("%d", &numbers[i]);

for(int i = 0; i < 14; i++){

if (numbers[i] > 0) {

positive += 1;

suma = suma + numbers[i];

}

}

printf("\nКількість додатних чисел = %d\nСума додатних чисел = %d", positive, suma);

return 0;

}

Висновок:

Починаючи виконання лабораторної роботи згідно порядку виконання лабораторної роботи та методичних вказівок, за першими пунктами перевірила свій обліковий запис в GitHub. Завдяки тому, що раніше він вже був створений згідно обов’язкового домашнього завдання, вносити правки не було потреби.

Далі, отримавши у викладача завдання за варіантом, перейшла до виконання задачі 7.1. Я виконала аналіз вимог і постановку задачі, на основі чого розробила та задокументувала тест-сьют TS\_7\_1. Виконала процедурно-орієнтовну алгоритмізацію задачі 7.1 з врахуванням вимоги про програмне оброблення текстової інформації у UTF-8.

Під час реалізації алгоритму розв’язування задачі 7.1 мовою програмування С (С18), стикнулась з певними складнощами. Спочатку були спроби реалізувати за допомогою команди system("chcp 65001 & cls"), але спроби були марними.

У зв’язку з цим, працювала з індексами літер. Вдалось це зробити також не відразу. Перша спроба базувалась на таблиці з індексами символів кодування UTF-8, але українські літери, на яких робила перевірку не підтримувались. Далі була спроба вивести індекси літер наступним чином:

char qf = 'й';

printf("%c = %ld\n", qf, qf);

char mf = 'Й';

printf("%c = %ld\n", mf, mf);

char a = 'ї';

printf("%c = %ld\n", a, a);

char qq = 'у';

printf("%c = %ld\n", qq, qq);

char bq = 'е';

printf("%c = %ld\n", bq, bq);

char fg = 'і';

printf("%c = %ld\n", fg, fg);

char hg = 'а';

printf("%c = %ld\n", hg, hg);

char yh = 'о';

printf("%c = %ld\n", yh, yh);

char rh = 'є';

printf("%c = %ld\n", rh, rh);

char rg = 'я';

printf("%c = %ld\n", rg, rg);

char er = 'и';

printf("%c = %ld\n", er, er);

char nm = 'ю';

printf("%c = %ld\n", nm, nm);

char bn = 'Ї';

printf("%c = %ld\n", bn, bn);

char hn = 'У';

printf("%c = %ld\n", hn, hn);

char ol = 'Е';

printf("%c = %ld\n", ol, ol);

char kl = 'І';

printf("%c = %ld\n", kl, kl);

char lk = 'А';

printf("%c = %ld\n", lk, lk);

char dc = 'О';

printf("%c = %ld\n", dc, dc);

char sc = 'Є';

printf("%c = %ld\n", sc, sc);

char sv = 'Я';

printf("%c = %ld\n", sv, sv);

char wb = 'И';

printf("%c = %ld\n", wb, wb);

char fw = 'Ю';

printf("%c = %ld\n", fw, fw);

Даний метод також був не вдалий, при спробі вивести літеру за допомогою індексу, отриманого вище, виводились інші символи.

Тоді я вирішила вивести сі символи за допомогою наступного циклу:

for (int i = 1; i < 256; i++){

printf("%c = %d\n", i, i);

}

В результаті, індекси відповідали табличним. Оскільки вже знаю, що вони не працюють, трішки змінила цикл:

for (int i = 0; i < -256; i++){

printf("%c = %d\n", i, i);

}

Використовуючи коди літер, отримані в результаті останнього циклу для літер кирилиці, програмна реалізація задачі 7.1 почала працювати.

Далі я приступила до реалізації задачі 7.2. виконала аналіз вимог і постановку задачі, на основі чого розробила та задокументувала тест-сьют TS\_7\_2. Виконала процедурно-орієнтовану алгоритмізацію задачі 7.2 та задокументувала його. Під час реалізації задачі 7.2 з труднощами не стикнулась