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 КБ-23

\_\_\_\_\_\_\_\_ Олексій КОНСТАНТИНОВ

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

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

Кропивницький – 2023

**ТЕМА: ПРОГРАМНА РЕАЛІЗАЦІЯ ОБРОБЛЕННЯ МАСИВІВ ДАНИХ ТА СИМВОЛЬНОЇ ІНФОРМАЦІЇ ЗА СТАНДАРТОМ UNICODE**

**МЕТА:** набути ґрунтовних вмінь і практичних навичок синтезу алгоритмів оброблення масивів даних та символьної (текстової) інформації у кодуваннях UTF-8 і CP866, їх програмної реалізації мовою програмування мовою програмування С (ISO/IEC 9899:2018) задля реалізації програмних засобів у вільному кросплатформовому Code::Blocks IDE.

**ЗАВДАННЯ:**

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 5**

Задача 7.1. Користувач вводить речення, яке закінчується «.» або «!». Вивести кількість символів «є» у введеному реченні; якщо зазначений символ відсутній, вивести відповідне речення.

1) Аналіз задачі 7.1:

Вхідні дані: речення, що завершується «.» або «!»

Вихідні дані: кількість символів «є», за відсутності їх у реченні вивести відповідне повідомлення.

2) Алгоритм:

1. Запропонувати користувачу ввести речення, зазначити, що речення повинне закінчуватися «.» або «!».

2. Перевірити кожен елемент рядка. Якщо система знайшла «.» або «!», пункт 5.

3.Якщо у реченні зустрічається символ «є», збільшити показання лічильника.

4. Перехід на крок 2

5. Якщо значення лічильника більше 0 – вивести значення лічильника, інакше вивести відповідне повідомлення.3

3) Лістинг програми:

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

int main() {

char \*locale = setlocale(LC\_ALL, "");

char sent[1000];

int count\_e = 0;

printf("Введiть речення (закiнчiть крапкою або знаком оклику): ");

gets(sent);

for (int i = 0; sent[i] != '.' && sent[i] != '!'; i++) {

if (sent[i] == 'є'||sent[i]=='Є')

count\_e++;

}

if (count\_e > 0) {

printf("Кiлькiсть символiв 'є' у введеному реченнi: %d\n", count\_e);

} else {

printf("Символ 'є' вiдсутнiй у введеному реченнi.\n");

}

getch();

return 0;

}

Задача 7.2. Вхід: масив з 20 чисел.

Вихід: інформація про те, яких елементів більше : парних чи непарних за значенням.

1) Аналіз задачі 7.2:

Вхідні дані: масив з 20 елементів.

Вихідні дані: інформація про те, яких елементів більше : парних чи непарних за значенням.

2) Алгоритм:

1. Запропонувати користувачу заповнити масив з 20 натуральних чисел.

2. Перевірити кожен елемент масиву. Якщо цикл спрацював 20 разів, пункт 5.

3. Якщо залишок від ділення на два дорівнює нулю, збільшити лічильник парних елементів. Інакше, збільшити лічильник непарних елементів.

4. Перехід на крок 2.

5. Якщо значення лічильника з парними числами більше, вивести відповідне повідомлення. Інакше, вивести повідомлення, яке вказує, що непарних чисел більше.

3) Лістинг програми:

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

int main()

{

srand(time(NULL));

const int SIZE = 20;

int odd\_num = 0, even\_num = 0;

int arr[SIZE];

printf("Enter 20 natural numbers: ");

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

arr[i]= 1 + rand()%(30);

printf("%d ",arr[i]);

if(arr[i]%2==0) even\_num++;

else odd\_num++;

}

printf("\n");

if(even\_num>odd\_num) printf("There are more even numbers than odd numbers");

else ("There are more odd numbers than even numbers");

getch();

}

Висновок. У ході лабораторної роботи, я реалізував програмне забезпечення для задачі 7.1, у якому використав цикл для перевірки кожного елемента рядка та операцію розгалуження для реалізації виводу різних повідомлень, в залежності від отриманих результатів. Також я реалізував програмне забезпечення для завдання 7.2 у якому я використав цикл for для перевірки кожного елемента масиву і оператор розгалуження if для реалізації підрахунку парних та непарних елементів масиву й виводу відповідного повідомлення, в залежності від отриманих результатів.