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 КН-23

Царенко Станіслав

ПЕРЕВІРИВ

викладач кафедри кібербезпеки

та програмного забезпечення

Ганна ДРЄЄВА

Кропивницький – 2024

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

**МЕТА:** полягає у набутті 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.

**ПОРЯДОК ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ ТА МЕТОДИЧНІ РЕКОМЕНДАЦІЇ:**

1. У веббраузері перейти за посиланням https://github.com/ та зареєструватись на вебсервіс системи контролю версій GitHub.

2. Після створення облікового запису (account) і підтвердження реєстрації через електронну пошту, у Personal settings заповнити Public profile:

Name: власне ім’я й прізвище латиницею,

Profile picture: офіційне фото (ділового стилю),

URL: www.kntu.kr.ua,

Company: Central Ukrainian National Technical University,

Location: Ukraine, Kropyvnytskyi.

3. Продемонструвати викладачеві створений обліковий запис, після чого надіслати GitHub username на [dorenskyiop@kntu.kr.ua](mailto:dorenskyiop@kntu.kr.ua).

4. Отримати у викладача завдання за варіантом, задокументувати мету лабораторної роботи, завдання, номер варіанту.

5. На одному з локальних дисків ПК (D:, E: чи ін.) створити теку BMTP-LAB7-прізвище, в ній ― теки prj, Software та TestSuite.

6. Виконати аналіз умови і постановку задачі 7.1, встановлених вимог до програмного засобу, який розроблятиметься, на основі чого розробити й задокументувати тест-сьют TS\_7\_1, який повинен складатись з не менш як десяти тест-кейсів.

7. Виконати процедурно-орієнтовану алгоритмізацію задачі 7.1 з врахуванням вимоги про програмне оброблення текстової інформації у UTF-8, при чому кодування вхідних і вихідних даних залежить від джерела й приймача інформації відповідно; здобутий алгоритм розв’язування задачі 7.1 задокументувати.

8. В Code::Blocks IDE створити новий проект консольного застосунку з назвою прізвище-task\_7\_1 у /prj.

9. Алгоритм розв’язування задачі 7.1 реалізувати мовою програмування С (С18), проект прізвище-task\_7\_1 закрити.

10. З теки проекта прізвище-task\_7\_1 за допомогою "Блокнот" відкрити файл main.c.

11. Перезберегти файл main.c у UTF-8 (Файл → Зберегти як → Кодування → UTF-8 → Зберегти).

12. Закрити файл main.c. Базові методології та технології програмування

13. В Code::Blocks IDE відкрити прізвище-task\_7\_1, скомпілювати і закрити проект; створений у /prj файл програми прізвищеtask\_7\_1.ехе скопіювати у /BMTP-LAB7-прізвище/Software.

14. За допомогою тестового набору TS\_7\_1 із теки /TestSuite, виконати системне тестування прізвище-task\_7\_1.ехе, який знаходиться у теці /BMTP-LAB7-прізвище/Software; отриманий тестовий артефакт — заповнений тест-сьют TS\_7\_1 — зберегти та долучити до звіту як додаток.

15. За умови, що статус всіх тест-кейсів TS\_7\_1 є passed, вихідний код (лістинг) проекту прізвище-task\_7\_1 задокументувати; у іншому випадку слід виконати відлагодження запропонованого алгоритму і/або ПЗ, проект прізвище-task\_7\_1 скомпілювати, системне тестування додатка прізвище-task\_7\_1.ехе повторити, результати повторної реалізації й тестування задокументувати.

16. Виконати аналіз умови і постановку задачі 7.2, встановлених вимог до програмного засобу, який розроблятиметься, на основі чого розробити й задокументувати тест-сьют TS\_7\_2, який повинен складатись з не менш як десяти тест-кейсів.

17. Виконати процедурно-орієнтовану алгоритмізацію задачі 7.2, отриманий алгоритм розв’язування задачі 7.2 задокументувати.

18. В Code::Blocks IDE створити новий проект консольного застосунку з назвою прізвище-task\_7\_2 у /prj.

19. Алгоритм розв’язування задачі 7.2 реалізувати мовою програмування С (С18), скомпілювати проект; створений файл програми прізвище-task\_7\_2.ехе скопіювати у /Software. 20. За допомогою TS\_7\_2 із /TestSuite, виконати системне тестування /BMTP-LAB2-прізвище/Software/прізвище-task\_7\_2.ехе; заповнений TS\_7\_2 долучити до звіту як додаток.

21. За умови, що статус всіх тест-кейсів TS\_7\_2 є passed, вихідний код (лістинг) проекту прізвище-task\_7\_2 задокументувати; у іншому випадку слід виконати відлагодження запропонованого алгоритму і/або ПЗ, проект прізвище-task\_7\_2 скомпілювати, системне тестування додатка прізвище-task\_7\_2.ехе повторити, результати повторної реалізації й тестування задокументувати.

22. Одержані результати виконання завдань лабораторної роботи (/BMTP-LAB7-прізвище) зберегти на флеш-накопичувач, у хмарному сховищі даних тощо.

23. Проаналізувати хід виконання завдань і одержані результати, сформулювати обґрунтовані висновки та викласти їх обсягом не менше двох сторінок машинного (комп’ютерного) тексту; також слід зазначити особисті враження від виконання лабораторної роботи, викласти вмотивовані пропозиції, обґрунтовані зауваження, конструктивну критику16, рекомендації тощо.

24. Інсталювати Git (Git Bash) https://git-scm.com/book/uk/v2/ВступІнсталяція-Git, використовуючи “Pro Git” <https://gitscm.com/book/uk/v2>.

25. На github.com/student-name створити Git-репозиторій (New → Create a new repository) та за допомогою Git (Git Bash) завантажити у нього теку BMTP-LAB7-прізвище (git init, git add, git commit, git branch, git remote, git push; https://techglimpse.com/git-push-github-token-based-passwordless/), а також звіт, оформлений згідно з ДСТУ 3008:2015 зі структурою й змістом, визначеними цим порядком виконання роботи.

26. Додати викладача і одногрупників до спільної роботи над проєктом BMTP-LAB7-прізвище: Settings → Access → Collaborators → Add people → Add NAME to REPOSITORY (NAME викладача — на дошці в аудиторії, у якій проходить заняття).

27. У комітах Git-репозиторію BMTP-LAB7-прізвище (https://github.com/ . . . /commits/main/) відреагувати і/або відповісти на коментарі співучасників ІТ-проєкту ( Commits), захистити результати й здобути зарахування роботи викладачем.

Варіант 6

Задача 7.1

Ввести рядок українською мовою. Вивести кількість символів ‘Ж’, якщо немає, вивести відповідне повідомлення.

#include <stdio.h>

#include <string.h>

int main() {

system("chcp 65001 & cls");

char sentence[1000];

int count = 0;

printf("Введіть речення: ");

scanf("%[^\n]s", sentence);

int len = strlen(sentence);

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

if (sentence[i] == 'Ж' || sentence[i] == 'ж') {

count++;

}

}

if (count > 0) {

printf("Кількість 'Ж' в реченні: %d\n", count);

} else {

printf("В цьому реченні немає 'Ж'\n");

}

return 0;

}

ПОЧАТОК

1. Введення sentence.

2. Проходження циклу і перевірка умови.

3. Якщо count > 0, виведення кількість 'Ж' в реченні, інакше виведення відповідного повідомлення.

КІНЕЦЬ

Задача 7.2

Ввести 15 дійсних чисел. Вивести значення яких є більше першого і менше останнього.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

void tag() {

printf("┌──────────────┬────────────────────────┐\n");

printf("│ Виконав: │ Царенко Станіслав │\n");

printf("└──────────────┴────────────────────────┘\n");

}

int main() {

system("chcp 65001 & cls");

tag();

const int len = 15;

int arr[len];

int count = 0;

printf("Введіть %d цифр: \n", len);

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

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

}

printf("\nМасив: \n");

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

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

if (arr[i] > arr[0] && arr[i] < arr[len - 1])

count++;

}

printf("\nКількість елементів більше першого і менше останнього: %d", count);

return 0;

}

ПОЧАТОК

1. Проходження циклу і введення arr[i].

2. Проходження циклу, виведення масиву arr[] і перевірка умови.

3. Виведення кількості елементів які більше першого і менше останнього елемента.

КІНЕЦЬ

**Висновки:**

По-перше, для виконання цієї лабораторної роботи треба реєстрація на GitHub та створення облікового запису: щоб мати можливість працювати над проектом та спільною роботою з колегами.

Далі потрібно проаналізувати умови завдань 7.1 і 7.2: Обидві задачі вимагають оброблення введених даних і виведення результатів за певними умовами. Задача 7.1 стосується рядків українською мовою, в той час як 7.2 - це робота з масивами чисел.

Слід за цим йде програмна реалізація задач 7.1 і 7.2: Обидва завдання реалізовані у мові програмування C, з використанням відповідних конструкцій мови для оброблення даних та виведення результатів.

Після реалізації кожної задачі було виконане системне тестування за допомогою відповідних тест-сьютів. Результати тестування показали, що програми працюють вірно і відповідають вимогам.

Після завершення завдань було створено Git-репозиторій на GitHub для спільної роботи над проектом з усіма учасниками, що дозволить зручно контролювати та оцінити пророблену роботу.

Отже, робота над проектом була успішною, дозволила отримати практичні навички 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.

А також роботі в GitHub. Важливою частиною був аналіз результатів та формулювання висновків для подальшого вдосконалення навичок та процесу роботи.