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

Турлюн Влад

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.

*Задача 7.1*

*#include<iostream>*

*using namespace std;*

*int main (){*

*int x,y;*

*cout << "INPUT x->";*

*cin >> x;*

*cout << "INPUT y->";*

*cin >> y;*

*for (int c=0;c<y;c++){*

*for (int i=0;i<x;i++)*

*cout<<"\*";*

*cout<<endl;*

*}*

*return 0;*

*}*

1. **Початок**
2. **Введення x**
3. **Введення y**
4. **Цикл c від 0 до y-1**
   * **Цикл i від 0 до x-1**
     + **Виведення "**\*"\*\*
   * **Кінець циклу i**
   * **Виведення нового рядка**
5. **Кінець циклу c**
6. **Кінець**

*Задача 7.2*

*#include<iostream>*

*#include<stdio.h>*

*int main (){*

*int i, S1=0, S3=0, S157=0, S256=0, S1000=0, a[10];*

*//cout<<"enter 10 numbers"<<endl;*

*printf("enter 10 numbers");*

*for (i=0; i<10; i++)*

*{*

*/\*cout<<"a["<< i+1 << "]=";*

*cin>>a[i];\*/*

*printf("a[%d] =", i+1);*

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

*if (a[i] ==1) S1++;*

*if (a[i] ==3) S3++;*

*if (a[i] ==157) S157++;*

*if (a[i] ==256) S256++;*

*if (a[i] ==1000) S1000++;*

*}*

*/\*cout<<"S1="<<S1;*

*cout<<"S3="<<S3;*

*cout<<"S157="<<S157;*

*cout<<"S256="<<S256;*

*cout<<"S1000="<<S1000;\*/*

*printf("\n 1= %d, 3= %d, 157= %d, 256= %d, 1000= %d", S1,S3,S157,S256,S1000);*

*return 0;*

*}*

1. **Початок**
2. **Ініціалізація змінних**: i, S1, S3, S157, S256, S1000.
3. **Введення чисел**:
   * Цикл for для 10 чисел:
     + **Введення числа**:
       - Виведення тексту "a[i] ="
       - Введення значення числа a[i].
       - Перевірка числа:
         * Якщо a[i] рівне 1, збільшуємо S1 на 1.
         * Якщо a[i] рівне 3, збільшуємо S3 на 1.
         * Якщо a[i] рівне 157, збільшуємо S157 на 1.
         * Якщо a[i] рівне 256, збільшуємо S256 на 1.
         * Якщо a[i] рівне 1000, збільшуємо S1000 на 1.
4. **Виведення результатів**:
   * Виведення кількості чисел, рівних 1, 3, 157, 256 та 1000.
5. **Кінець**

Завдання 7.1

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | x=3,y=4 | \*\*\*  \*\*\*  \*\*\*  \*\*\* | Passed |
| 2 | x = 5, y = 3 | \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\* | Passed |
| 3 | X=7,y=4 | \*\*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\*  \*\*\*\*\*\*\* | Passed |

Завдання 7.2

|  |  |  |  |
| --- | --- | --- | --- |
| Ід-р тест-кейса / Test Case ID | Дії (кроки) /  Action  (Test Steps) | Очікуваний  результат /  Expected Result | Результат тестування (пройшов/не вдалося/ заблокований) /  Test Result (passed/failed/ blocked) |
| 1 | 1 1 1 1 1 1 1 1 1 1 | 1=10, 3=0, 157=0, 256=0, 1000=0 | Passed |
| 2 | 3 3 3 3 3 3 3 3 3 3 | 1=0, 3=10, 157=0, 256=0, 1000=0 | Passed |
| 3 | 157 157 157 157 157 157 157 157 157 157 | 1=0, 3=0, 157=10, 256=0, 1000=0 | Passed |
| 4 | 256 256 256 256 256 256 256 256 256 256 | 1=0, 3=0, 157=0, 256=10, 1000=0 | Passed |
| 5 | 1000 1000 1000 1000 1000 1000 1000 1000 1000 1000 | 1=0, 3=0, 157=0, 256=0, 1000=10 | Passed |

Висновок: мета лабораторної роботи полягає в набутті вмінь і навичок розв'язання задач, використовуючи мову програмування С і С++, у вивченні основних процедур Git.