Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

з лабораторної роботи № 7

з дисципліни “Базові методології та технології програмування”

на тему

“Програмна реалізація оброблення масивів даних та символьної інформації”

Виконав студент

академічної групи КІ-20

Микитенко Д. Ю.

Перевірив старший викладач

Поліщук Л. І.

Кропивницький-2021

Мета роботи 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 до лабораторної роботи:

1. Створити програмне забезпечення для задачі 7-1

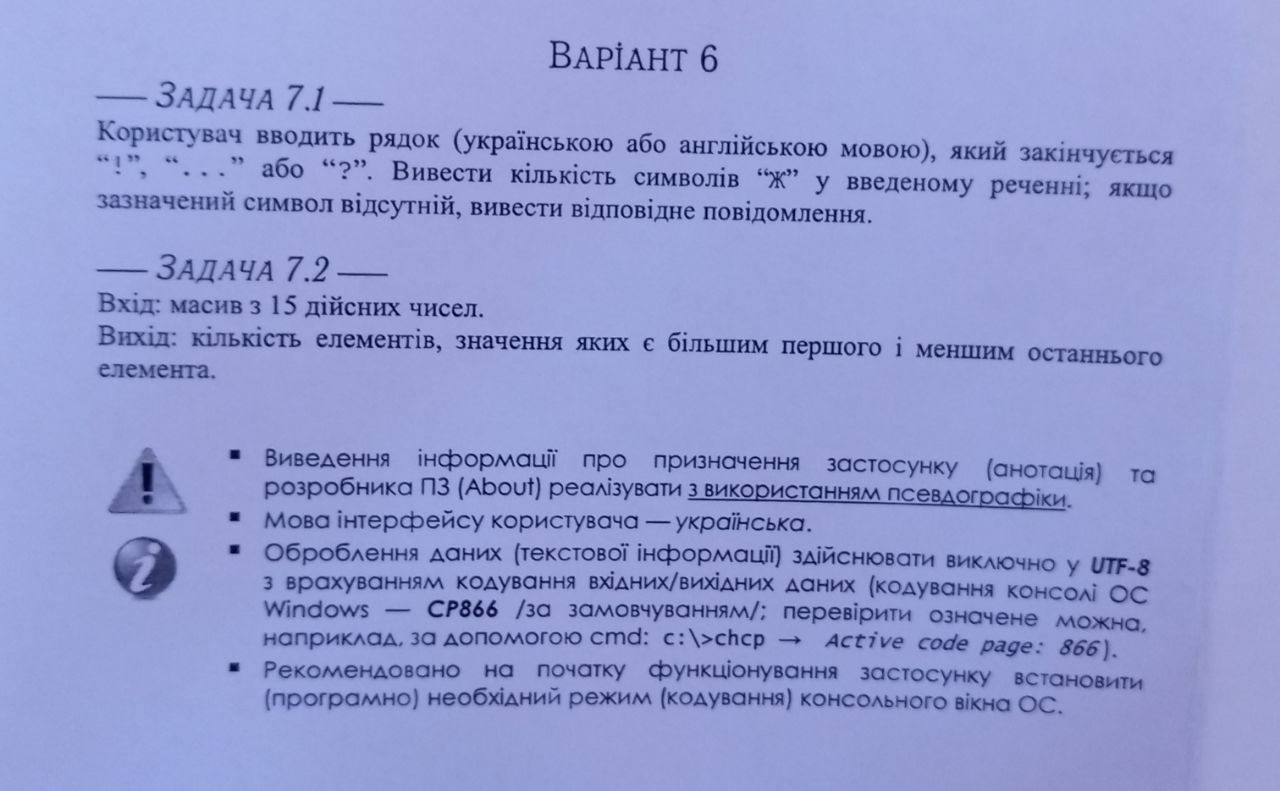
2. Реалізувати програмне забезпечення розв’язування задачі 7.2. 

Рисунок 1 – Завдання роботи варіант № 6

1. Згідно з п.3-5 Порядку були створені деки: prj, Software, TestSuite.
2. За допомогою аналізу був сформований перелік змінних(вхідні та

вихідні значення) для задачі 7.1.

Вихідні дані:

1. Amount – кількість літер Ж у реченні.

Вхідні дані:

1. Amount – ціле додатне число, не менше 0.
2. Str[100] – речення.

Обмеження:

1. Кількість літер Ж у реченні не може бути меншою за 0.
2. Результат не повинен бути більшим за 32767.

Математична модель:

1. Якщо у реченні зустрічається літера Ж, то amount = amount + 1;

Програмна реалізація задачі 7.1 була створена у IDE Code::Blocks у папці prj.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <wchar.h>

#include <string.h>

#include <ctype.h>

#include <io.h>

#include <fcntl.h>

wchar\_t str[100];

int amount = 0;

void localisation ()

{

system("chcp 866");

\_setmode(\_fileno(stdout),\_O\_U16TEXT);

\_setmode(\_fileno(stdin),\_O\_U16TEXT);

}

void lettersCounter()

{

for(int i = 0; i < wcslen(str); i++)

{

if(str[i] == L'ж' || str[i] == L'Ж' || str[i] == 'j' || str[i] == 'J')

{

amount++;

}

}

}

void output()

{

wprintf(L"\nКiлькiсть літер Ж у реченi - ");

wprintf(L"%d\n", amount);

}

void emptyOutput ()

{

wprintf(L"\nРечення не містить зазначенних літер.\n");

}

int main()

{

localisation();

for(int i = 0; i < 80; i++)

{

wprintf(L"▄");

}

wprintf(L"\nУ даннiй програмi користувач вводить речення,яке закiнчується на '...', '!', '?'.\

\nА програма повинна вивести кiлькiсть літер Ж у введеному реченi,\nякщо цих символiв немає, то вивести відповідне повідомлення.\n");

for(int i = 0; i < 80; i++)

{

wprintf(L"▄");

}

wprintf(L"\nВерсiя 1.0\nРозробник: Микитенко Даниїл Юрійович.\nЦНТУ, 2021 рiк.\n");

wprintf(L"\nУведiть речення:");

fgetws ( str, 100, stdin );

lettersCounter();

if(amount == 0)

{

emptyOutput();

return 0;

}

output();

}

2) Вихідні дані:

1. Amount – кількість чисел, що задовільняют умову.

Вхідні дані:

1. Amount – ціле додатне число, не менше 0.
2. Arr[15] – масив чисел

Обмеження:

1. Кількість чисел, що задовільняют умову, не може бути меншим 0.
2. Результат не повинен бути більшим за 32767.

Математична модель:

1. Якщо число більше за перший елемент масиву та менше за останній елемент массиву, amount = amount + 1;

Програмна реалізація задачі 7.2 була створена у IDE Code::Blocks у папці prj.

#include <stdio.h>

#include <stdlib.h>

#include <locale.h>

#include <wchar.h>

#include <string.h>

#include <ctype.h>

#include <io.h>

#include <fcntl.h>

int main()

{

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

for(int i = 0; i < 80; i++)

{

printf("-");

}

printf("\nУ даннiй програмi користувач вводить 15 чисел.\

\nА програма повинна вивести кiлькiсть чисел, не рівних 0, які більші за перший елемент массиву та менше за останній.\n");

for(int i = 0; i < 80; i++)

{

printf("-");

}

printf("\nВерсiя 1.0\nРозробник: Микитенко Даниїл Юрійович.\nЦНТУ, 2021 рiк.\n");

int arr[15] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

for(int i = 0; i < 15; i++)

{

printf("Введіть число %d: ", i+1);

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

if(arr[i] == 0)

{

printf("Неправильні вхідні дані.");

return 0;

}

}

for(int i = 0; i < 15; i++)

{

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

}

int amount = 0;

for(int i = 0; i < 15; i++)

{

if(arr[i] > arr[0] && arr[i] < arr[14])

amount++;

}

printf("\nКількість елементів задовільняючих умову - %d", amount);

}

Висновок: у ході виконання лабораторної роботи № 7 треба було розробити програмну реалізацію завдання 7.1 та розробити програмну реалізацію завдання 7.2.

Щоб виконати перше завдання потрібно було ознайомитися з багатьма бібліотеками, а саме <wchar.h>, <ctype.h>, <io.h> та бібліотекою <fcntl.h>. Ознайомитися з принципом роботи з двухбайтовими символами. Ознайомитися з принципом роботи з текстом у шифровці UTF-8. Після цього був створений шаблон програмного забезпечення до задачі 7.1, адже ПЗ не було протестовано. Також потрібно було прослідити за тим, щоб машинний код алгоритму, написаний мовою С, відповідав вимогам, тобто: містив вичерпну інформацію про його призначення (анотація), розробника, контактні дані, версію й рік розробки щодо. Потім було створено 10 тест-сьютів для умови задачі, 2 з яких були тест-сьютами щодо інтерфейсу й вимог, й було перевірено кожен з них для нашого програмного забезпечення, після чого виявилося, що всі тест-сьюти пройшли тестування.

Труднощі, які виникли під час роботи, були підмічені, опрацьовані та виправлені. В основному це були труднощі саме з працею з двухбайтовими символами, адже для них все працює інакше. Після успішної компіляції та проходження тест-сьюту, .exe файл був перенесений з теки .prj до теки SoftWare, що була створена на самому початку виконання лабораторної роботи на подобі до лабораторних робіт №4-6.

Задля виконання другого завдання потрібно було уважно проаналізувати дану задачу, задокументувати вхідні та вихідні значення, обмеження, їх типи та побудувати математичну модель. Після цього був створений шаблон програмного забезпечення до задачі 7.2, адже ПЗ не було протестовано. Також потрібно було прослідити за тим, щоб машинний код алгоритму, написаний мовою С, відповідав вимогам, тобто: містив вичерпну інформацію про його призначення (анотація), розробника, контактні дані, версію й рік розробки щодо. Потім було створено 10 тест-сьютів для умови задачі, 2 з яких були тест-сьютами щодо інтерфейсу й вимог, й було перевірено кожен з них для нашого програмного забезпечення, після чого виявилося, що всі тест-сьюти пройшли тестування.

Підбиваючи підсумки цієї лабораторної роботи, були закріплені практичні вміння праці з тестуванням ПЗ, закріпленні знання з теми повторень(циклів), а також нових вимог до інтерфейсу програми. Також були здобуті практичні вміння з розробкою ПЗ для окремої задачі та створення інтерфейсу до неї, а також шифрування на базі UTF-8.

Всі труднощі, що виникли під час розробки програмного забезпечення, були опрацьовані.

Також новим було вивчення нових бібліотек, а саме:

* <wchar.h>
* <ctype.h>
* <io.h>
* <fcntl.h>

Новим була праця з двухбайтовими символами, адже для них застосовуються інші оператори та інші принципи роботи з обробкою даних. Вони були схожі до тих, що застосовувались раніше, але найбільша проблема була з заданням формату строки. Для того, щоб вказати, що строка належить до Юнікоду, потрібно було проставити літеру L, перед викликом строки. Також можна використовувати інший макрос - \_T. Він визначає чи принадлежить строка до Юнікоду й підставляє макрос L за потреби.