

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

«Програмування ч.1»

Звіт з лабораторної роботи №14
Тема: «Взаємодія з файлами»

Виконав:
ст. гр. КІТ-120А
Старовойтов Н.А.

Перевірив:
Челак В.В.

Харків – 2021

Мета: Отримати навички взаємодії з файлами та її використанням при розв'язуванні задач.

Індивідуальне завдання

Робота на оцінку “відмінно”.

1) Програму, яка була розроблена у попередній лабораторній роботі (робота з строками), змінити так, щоб:

- початкові дані вводилися з файлу;
- видача результуючих даних проводилася і в консоль, і у файл.
- ім'я вхідного та вихідного файлу було отримано від користувача;
- при запиті даних, користувач отримав повідомлення, що від нього очікують.

Завдання №5: "Текст - це програма на мові C. Визначити, скільки в ньому операторів циклу".

2) Вивести структуру файлів та каталогів, як це робить утиліта Linux tree. Виклик функції system заборонено.

3) Визначити об'єм запитаного каталогу. Результат нормалізувати. Формат розміру:

- не більше ніж 3 знаки до коми;
- не більше ніж 2 знаки після коми;
- єдиний випадок, коли в чисельній частині числа може бути 0 - коли розмір дорівнює 0 байт.

Опис програми

♦ read_from_file()

```
char read_from_file ( char * string )
```

Функція читає з вхідного файлу текст та зберігає його в одну строку

Аргументи

string - строка із заданим текстом

Повертає

*string - покажчик на строку із прочитаним текстом

♦ write_to_file()

```
void write_to_file ( int result )
```

Функція записує у вихідний файл результат - кількість операторів циклу в заданому тексті

Аргументи

result - кількість операторів циклу в заданому тексті

Рисунок 1 — Функції read_from_file та write_to_file

♦ get_count_of_operators()

```
int get_count_of_operators ( char * string )
```

Функція підраховує, скільки в заданому тексті операторів циклу.

Аргументи

string - строка із заданим текстом

Повертає

Кількість операторів циклу в заданому тексті

♦ main()

```
int main ( )
```

Головна функція.

Послідовність дій:

1. Виділяємо пам'ять під строку з текстом
2. Зчитуємо з вхідного файлу текст до строки string
3. Змінній result присвоюємо значення кількості операторів циклу в заданому тексті за допомогою функції `get_count_of_operators`
4. Записуємо у вихідний файл результат - кількість операторів циклу в заданому тексті

Повертає

0 - код успішного виконання програми

Граф всіх викликів цієї функції:

Рисунок 2 — Функції get_count_of_operators та main

Схеми алгоритмів функцій

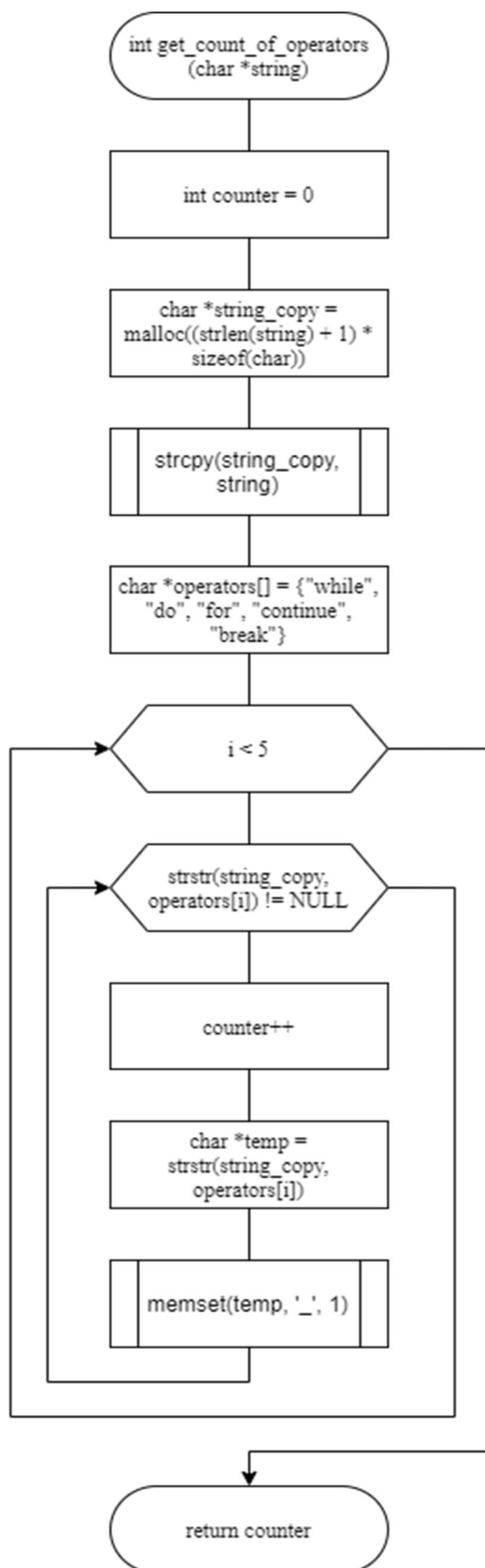


Рисунок 3 — Блок-схема функції `get_count_of_operator`

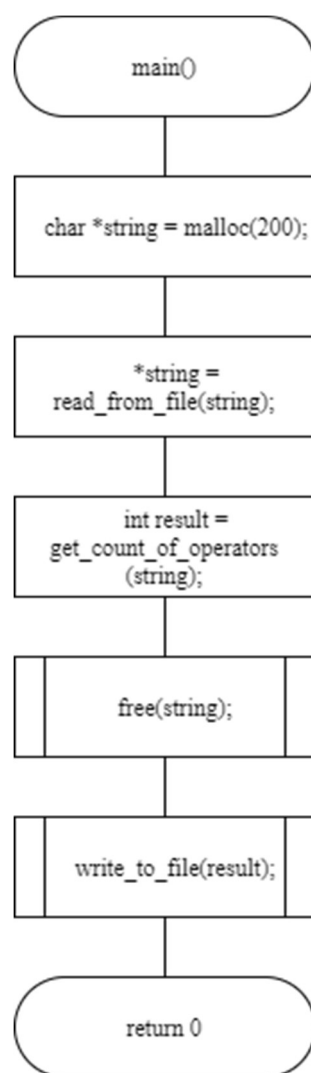


Рисунок 4 — Блок-схема функції `main` завдання 1

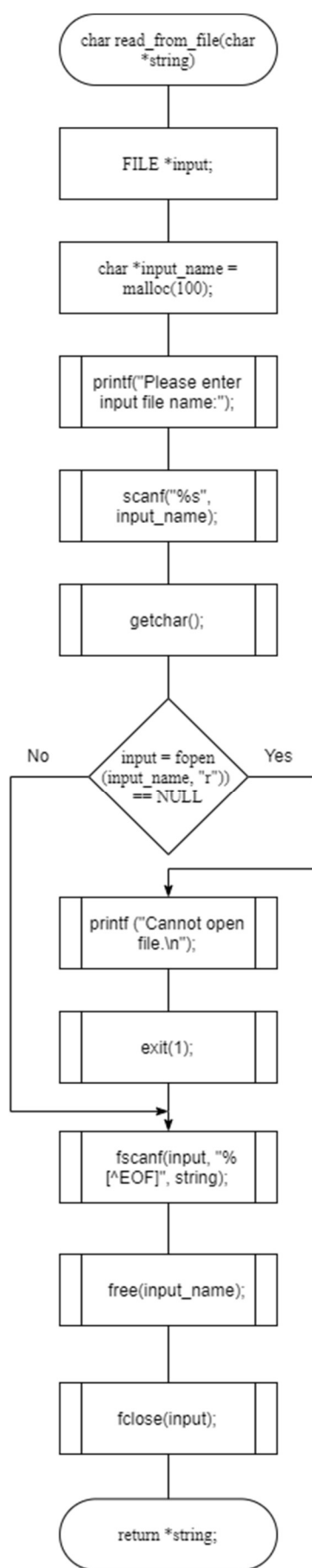


Рисунок 5 — Блок-схема функції read_from_file

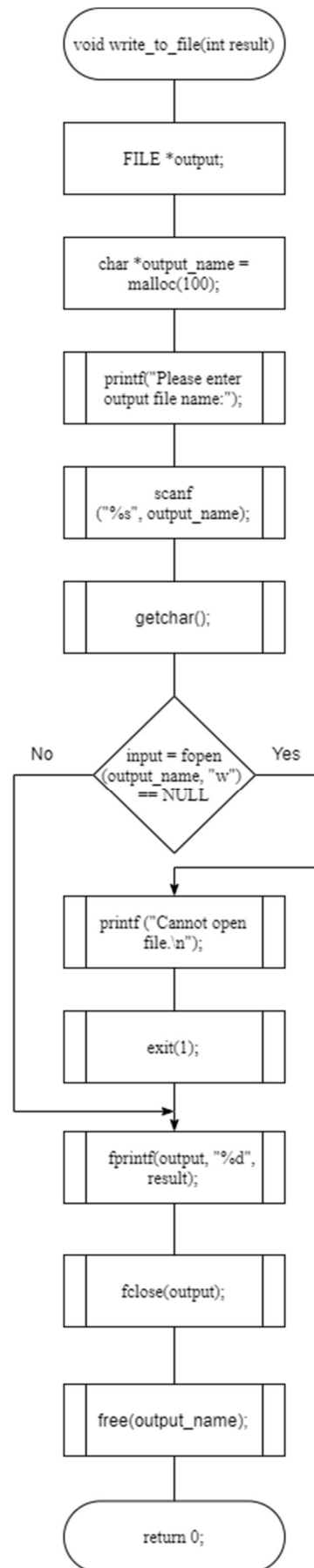


Рисунок 6 — Блок-схема функції write_to_file

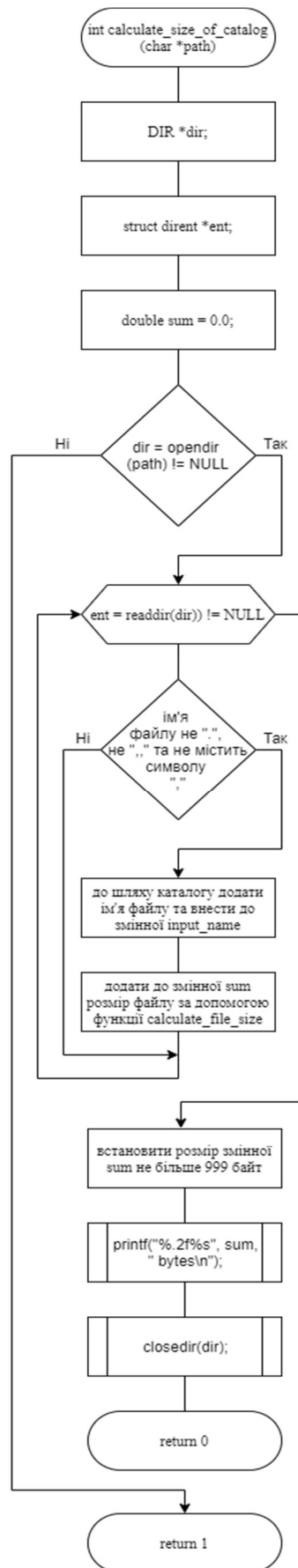


Рисунок 7 — Блок-схема функції calculate_size_of_catalog

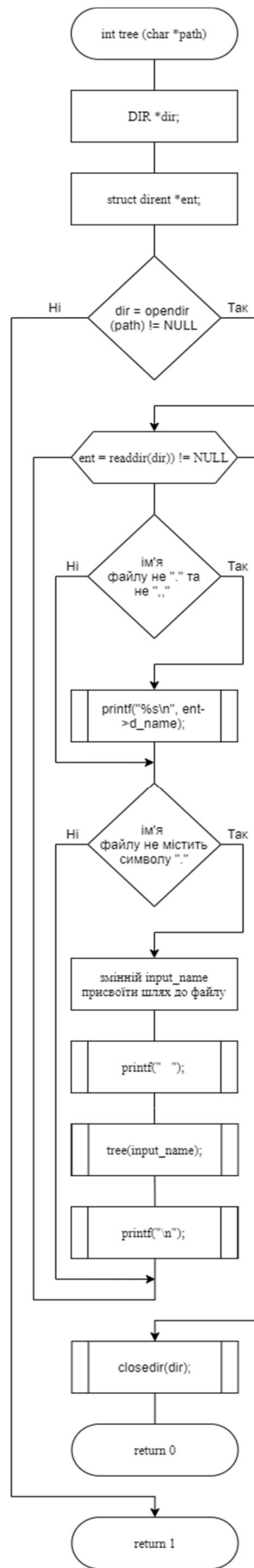


Рисунок 8 — Блок-схема функції tree

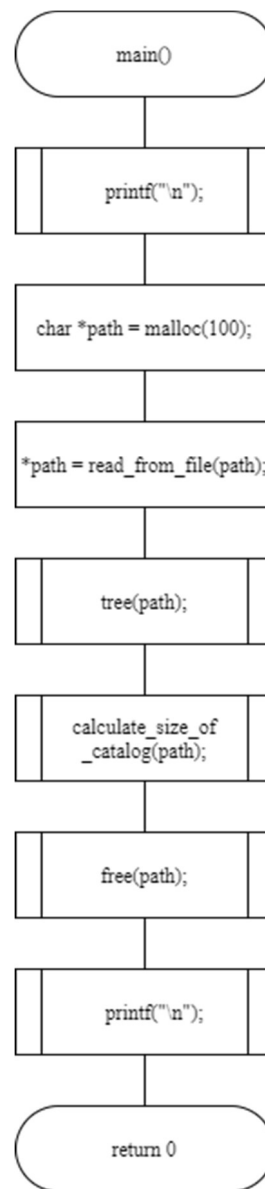
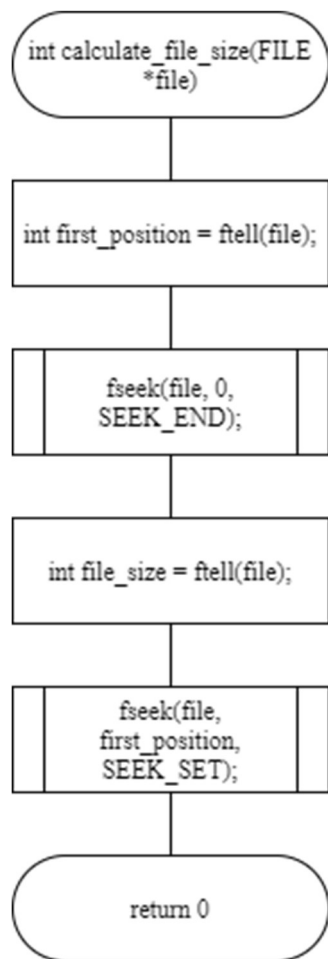


Рисунок 9, 10 — Блок-схема функції `calculate_file_size` та функції `main` другого та третього завдання

Текст програми

```

#include <stdlib.h>
#include <stdio.h>
#include <dirent.h>
#include <string.h>

char read_from_file(char *string) {
    FILE *input;
    char *input_name = malloc(100);
    printf("Please enter input file name:");
    scanf("%s", input_name);
    getchar();
    if ((input = fopen(input_name, "r")) == NULL) {
        printf("Cannot open file.\n");
        exit(1);
    }
}
  
```

```

    }
    fscanf(input, "%[^EOF]", string);
    free(input_name);
    fclose(input);
    return *string;
}

int calculate_file_size(FILE *file)
{
    int first_position = ftell(file);
    fseek(file, 0, SEEK_END);
    int file_size = ftell(file);
    fseek(file, first_position, SEEK_SET);
    return file_size;
}

int calculate_size_of_catalog (char *path) {
    DIR *dir;
    struct dirent *ent;
    printf("\n");
    double sum = 0.0;
    if ((dir = opendir(path)) != NULL) {
        /* print all the files and directories within directory */
        while ((ent = readdir(dir)) != NULL) {
            if (strchr(ent->d_name, '.') != NULL && strcmp(ent->d_name, ".") != 0 && strcmp(ent->d_name,
"..") != 0) {
                char *input_name = malloc(100);
                strcpy(input_name, path);
                strncat(input_name, "\\ ", 2);
                strncat(input_name, ent->d_name, sizeof(ent->d_name));

                FILE *file;
                if ((file = fopen(input_name, "r")) == NULL) {
                    printf("Cannot open file.\n");
                    exit(1);
                }
                sum += (double) calculate_file_size(file);
                free(input_name);
                fclose(file);
            }
        }

        if (sum > 999.0) sum = 999.0;
        printf("%.2f%s", sum, " bytes\n");
        closedir(dir);
        return 0;
    } else {
        perror("");
        return 1;
    }
}

int tree (char *path){
    DIR *dir;
    struct dirent *ent;
    if ((dir = opendir (path)) != NULL) {
        /* print all the files and directories within directory */
        while ((ent = readdir (dir)) != NULL) {
            if (strcmp(ent->d_name, ".") != 0 && strcmp(ent->d_name, "..") != 0) {
                printf("%s\n", ent->d_name);
            }
        }
    }
}

```

```

        if (strchr(ent->d_name, '.') == NULL) {
            char *input_name = malloc(100);
            strcpy(input_name, path);
            strncat(input_name, "\\", 2);
            strncat(input_name, ent->d_name, sizeof(ent->d_name));
            printf(" ");
            tree(input_name);
            printf("\n");
        }

    }
    closedir (dir);
    return 0;
} else {
    perror ("");
    return 1;
}
}

int main()
{
    printf("\n");

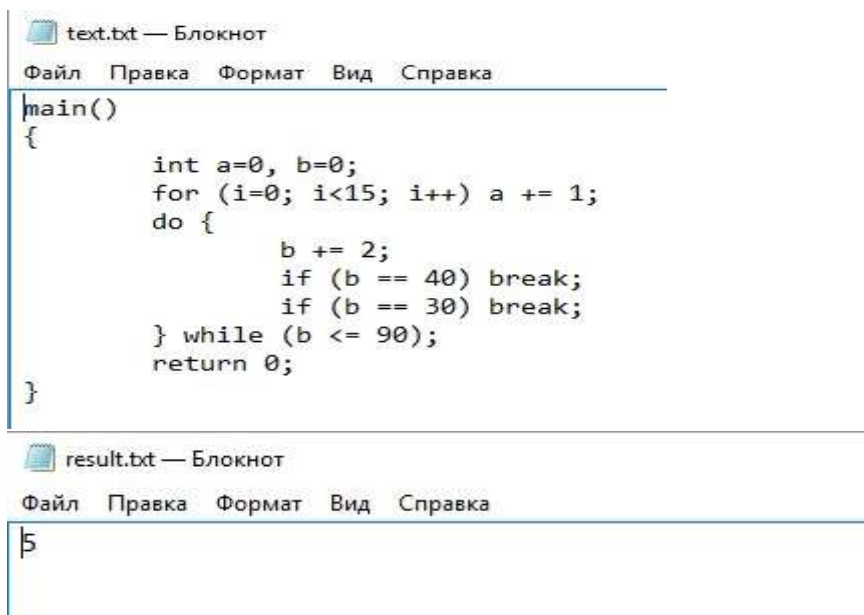
    char *path = malloc(100);
    *path = read_from_file(path);
    tree(path);
    calculate_size_of_catalog(path);

    printf("\n");

    return 0;
}

```

Результати роботи програми

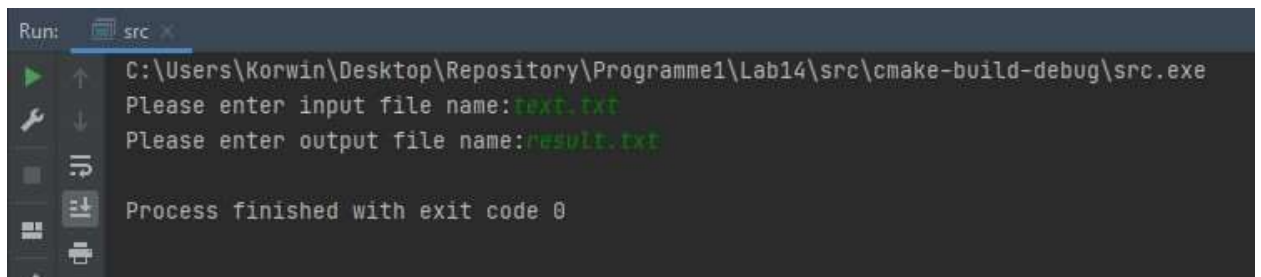


```

text.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
main()
{
    int a=0, b=0;
    for (i=0; i<15; i++) a += 1;
    do {
        b += 2;
        if (b == 40) break;
        if (b == 30) break;
    } while (b <= 90);
    return 0;
}

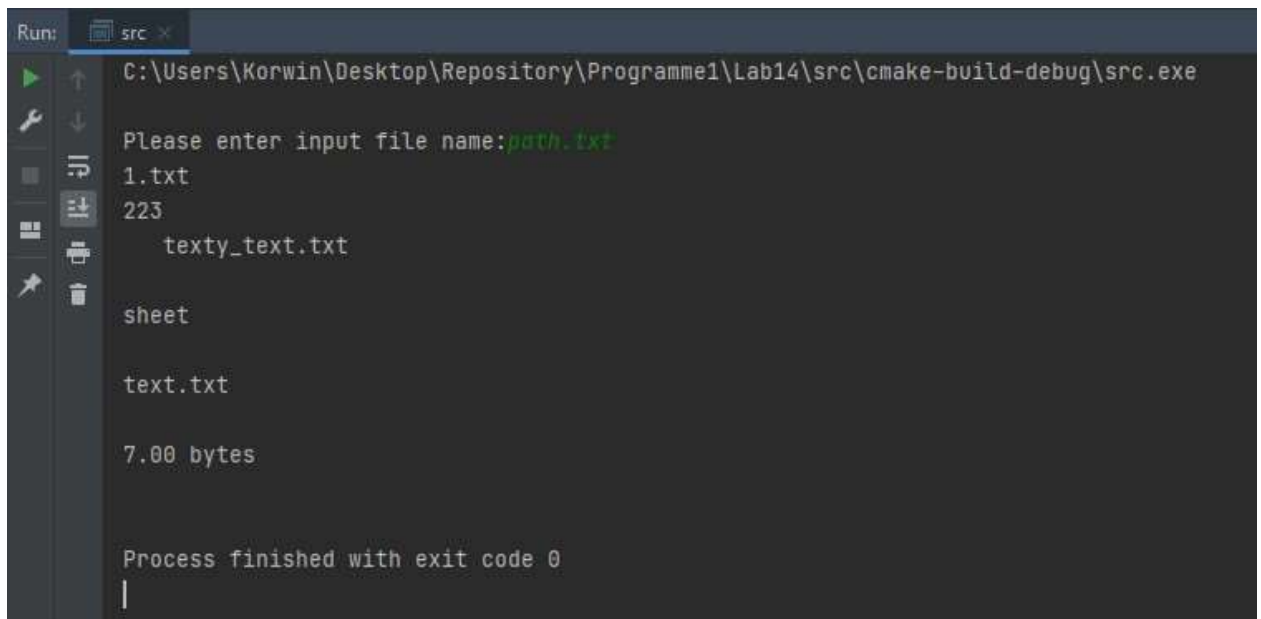
result.txt — Блокнот
Файл  Правка  Формат  Вид  Справка
5

```



```
Run: src x
C:\Users\Korwin\Desktop\Repository\Programme1\Lab14\src\cmake-build-debug\src.exe
Please enter input file name: text.txt
Please enter output file name: result.txt
Process finished with exit code 0
```

Рисунок 11 — Результат успішного виконання програми першого завдання



```
Run: src x
C:\Users\Korwin\Desktop\Repository\Programme1\Lab14\src\cmake-build-debug\src.exe
Please enter input file name: path.txt
1.txt
223
texty_text.txt
sheet
text.txt
7.00 bytes
Process finished with exit code 0
|
```

Рисунок 12 — Результат успішного виконання програми другого та третього завдання

Висновки

Під час виконання даної лабораторної роботи було отримано навички взаємодії з файлами та її використання при розв'язуванні задач.