

Міністерство освіти і науки України  
Інститут телекомунікацій, радіоелектроніки та електронної техніки  
**Звіт до лабораторної роботи № 17**

**Тема: «Дослідження способів організації потокового  
уведення/виведення в мові програмування C.»**

з дисципліни «Програмування частина 2»

Варіант № 6

виконав студент групи АП-11

Головацький Назар

перевірив доцент кафедри ТК

Чайковський І.Б

**Мета роботи:** Дослідження способів створення, оновлення та оброблення файлів потокового введення/виведення даних у мові С.

1. Дослідити та дати пояснення прикладів, викладених нижче.

A)

```
#include "stdio.h"
```

```
int main() {
```

```
    FILE *in; // Опис вказівника на файл
```

```
    int ch;
```

if ((in = fopen("proba", "r")) != NULL) { // Відкривається файл для читання, перевіряється чи він існує. Вказівник in тепер посилається на структуру FILE, що пов'язана із proba.

```
    while ((ch = getc(in)) != EOF) // Отримується символ із in
```

```
        putchar(ch, stdout); // Виведення символу в стандартний потік на екран.
```

```
    fclose(in); // Закриття файлу
```

```
    } else {
```

```
        printf("Файл proba не відкривається \n");
```

```
    }
```

```
    return 0;
```

```
}
```

Б)

```
#include <stdio.h>
```

```
int main() {
```

```
    FILE *ff;
```

```
    int base;
```

```
    ff = fopen("sam", "r"); // відкривається файл із іменем sam, який  
    ідентифікується зі вказівником на ff.
```

```
    fscanf(ff, "%d", &base); // Читається число з файлу sam і зберігається в  
    змінну base
```

```
    fclose(ff); // Закриття файлу sam
```

```
    ff = fopen("data", "a"); // відкривається файл data для доповнення
```

```
    fprintf(ff, "sam is %d.\n", base); // Запис у файл data
```

```
    fclose(ff); // Закриття файлу data
```

```
    return 0;
```

```
}
```

В)

```
#include <stdio.h>
```

```
#define LINE 80
```

```
int main() {
```

```
    FILE *ff;
```

```
    char string[LINE]; // Буфер для зчитування рядків
```

```
    ff = fopen("opus", "r"); // Відкриття файлу opus для читання
```

```
    while (fgets(string, LINE, ff) != NULL) // Зчитування рядків з файлу
```

```
        puts(string); // Виведення рядка на екран
```

```
    fclose(ff); // Закриття файлу
```

```
    return 0;
```

```
}
```

Г)

```
#include <stdlib.h>
```

```
#include <stdio.h>
```

```
int main() {  
    int f1, f2, f3, f4, f5;  
    FILE *fp;
```

```
    fp = fopen("C:\\temp\\sample.txt", "r"); // Відкриття файлу в режимі  
читання
```

```
    // Читання з файлу
```

```
    fscanf(fp, "%d\\n%d\\n%d\\n%d\\n%d\\n", &f1, &f2, &f3, &f4, &f5);
```

```
    printf("The values are %d, %d, %d, %d, %d \\n", f1, f2, f3, f4, f5); //
```

Виведення зчитаних значень

```
    fclose(fp); // Закриття файлу
```

```
    return 0;
```

```
}
```

Функції форматowanego обміну з файлами

fprintf() і fscanf()

fprintf()

```
int fprintf(FILE *stream, const char *format, ...);
```

Опис: Записує форматований текст у файл. Приклад:

```
FILE *fp = fopen("example.txt", "w");
```

```
int num = 10;
```

```
fprintf(fp, "The number is %d\n", num);
```

```
fclose(fp);
```

fscanf()

```
int fscanf(FILE *stream, const char *format, ...);
```

Опис: Зчитує форматований текст із файлу. Приклад:

```
FILE *fp = fopen("example.txt", "r");
```

```
int num;
```

```
fscanf(fp, "%d", &num);
```

```
printf("The number is %d\n", num);
```

```
fclose(fp);
```

Відмінності від printf() і scanf()

printf()

```
int printf(const char *format, ...);
```

Опис: Виводить форматований текст на екран. Приклад:

```
int num = 10;
```

```
printf("The number is %d\n", num);
```

scanf()

```
int scanf(const char *format, ...);
```

Опис: Зчитує форматований текст з клавіатури. Приклад:

```
int num;
```

```
scanf("%d", &num);
```

```
printf("The number is %d\n", num);
```

Ключові відмінності

Призначення:

fprintf() та fscanf() працюють з файлами.

printf() та scanf() працюють з консоллю.

Перший аргумент:

fprintf() і fscanf(): вказівник на файл (FILE \*).

printf() і scanf(): форматний рядок (const char \*).

3

```
#include <stdio.h>
```

```
int main(void) {
```

```
    FILE *pf;
```

```
    int k;
```

```
    // Відкриваємо файл для запису
```

```
    if ((pf = fopen("proba.txt", "w")) == NULL) {
```

```
        perror("proba.txt");
```

```
        return 1;
```

```
    }
```

```
    // Записуємо числа від 0 до 5 та їх куби в файл
```

```
    for (k = 0; k <= 5; k++) {
```

```
        fprintf(pf, "%d %d\n", k, k * k * k);
```

```
    }
```

```
    // Закриваємо файл
```

```
    fclose(pf);
```

```
    return 0;
```

```
}
```

## **програма для читання даних з файлу "proba.txt"**

```
#include <stdio.h>
```

```
int main(void) {
```

```
    FILE *pf;
```

```
    int n, nn, l;
```

```
    if ((pf = fopen("proba.txt", "r")) == NULL) {
```

```
        perror("proba.txt");
```

```
        return 1;
```

```
    }
```

```
    for (l = 0; l <= 5; l++) {
```

```
        fscanf(pf, "%d %d\n", &n, &nn);
```

```
        printf("Число: %d, Куб: %d\n", n, nn);
```

```
    }
```

```
    fclose(pf);
```

```
    return 0;
```

```
}
```

4

```
#include <stdio.h>
#include <stdlib.h>

int main() {
    FILE *file;
    int array[12] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12};
    int i;
    int read_array[12];

    // Записуємо масив у файл
    file = fopen("array.bin", "wb");
    if (file == NULL) {
        fprintf(stderr, "Помилка при відкритті файлу для запису.\n");
        return 1;
    }
    fwrite(array, sizeof(int), 12, file);
    fclose(file);

    // Читаємо масив з файлу
    file = fopen("array.bin", "rb");
    if (file == NULL) {
        fprintf(stderr, "Помилка при відкритті файлу для читання.\n");
        return 1;
    }
    fread(read_array, sizeof(int), 12, file);
    fclose(file);

    // Виводимо прочитані значення на екран
    printf("Прочитані значення з файлу:\n");
    for (i = 0; i < 12; i++) {
        printf("%d ", read_array[i]);
    }
    printf("\n");

    return 0;
}
```



**Результат:**

Прочитані значення з файлу:

1 2 3 4 5 6 7 8 9 10 11 12