Міністерство освіти та науки України

Інститут «Телекомунікації, радіоелектроніки та радіотехніки»



Звіт до лабораторної роботи № 19

З дисципліни «Програмування частина 2»

Дослідження способів організації потокового уведення/виведення в мові програмування С.

Виконав студент групи АП-12

Дроф’як Святослав

Перевірив викладач

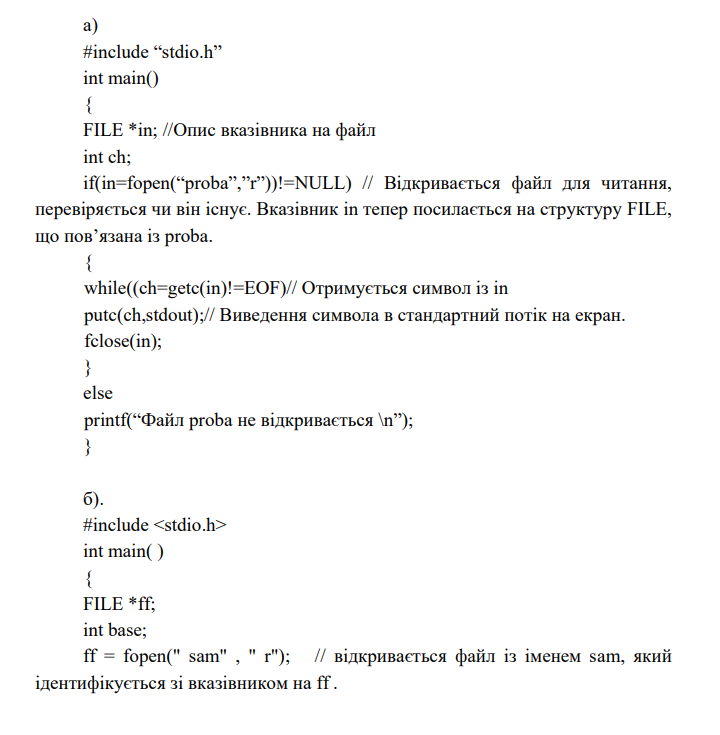
Пелех Н.В.

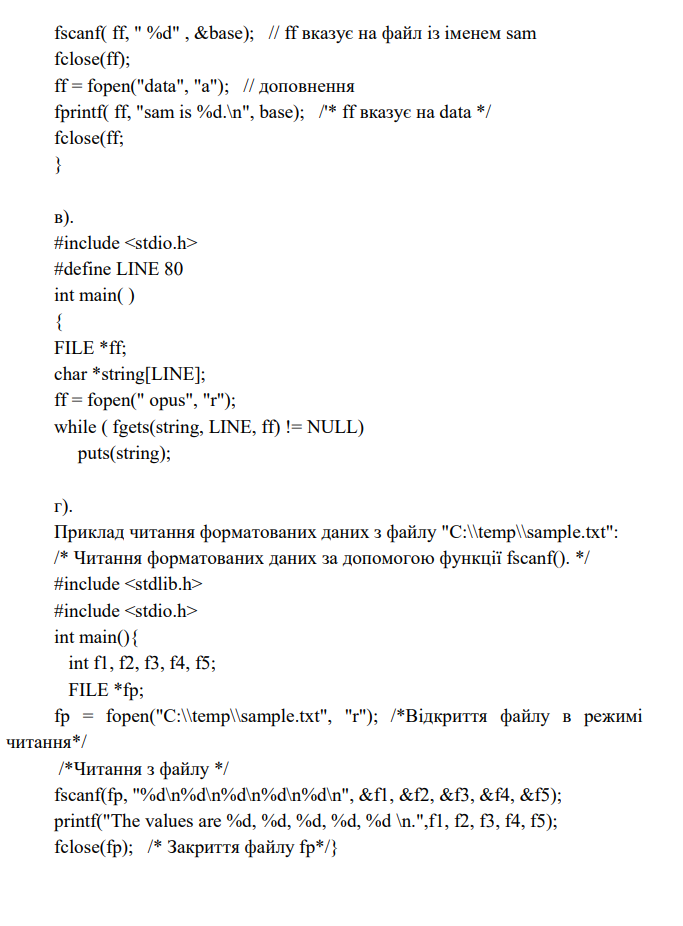
Львів 2024

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

Завдання:

1. Дослідити та дати пояснення прикладів, викладених нижче.





2.Розглянути функції форматного обміну з файлами fprintf(), fscanf() пояснити їх відмінності від функцій printf(), scanf().

3.Виконати програму, що створює файл proba.txt і записує в нього символьні зображення чисел від 0 до 5 і їх кубів. Наступною програмою прочитати дані із файлу proba.txt. У звіті дати детальне пояснення роботи програм

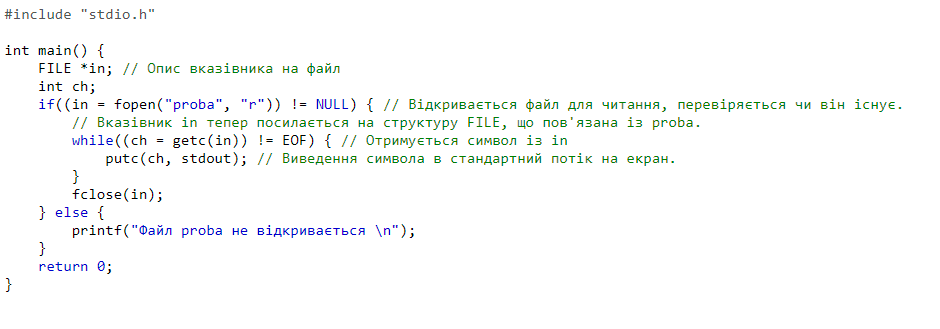
4. Виконати завдання згідно варіанта

|  |  |
| --- | --- |
| № вар. | Завдання |
| 2 | Задати 8 змінних цілого типу, записати їх в файл на диску, прочитати їх з файлу в масив розмірністю 8, масив вивести на екран. |

**Завдання 1**

**Реалізація:**

Програмна реалізація мовою С

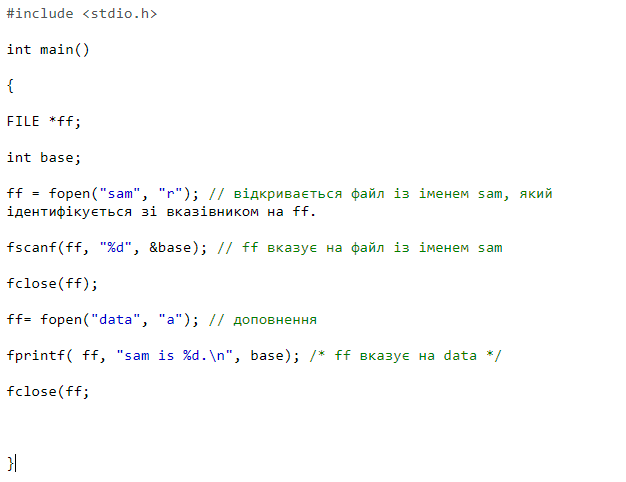


Код завдання 1, приклад a)



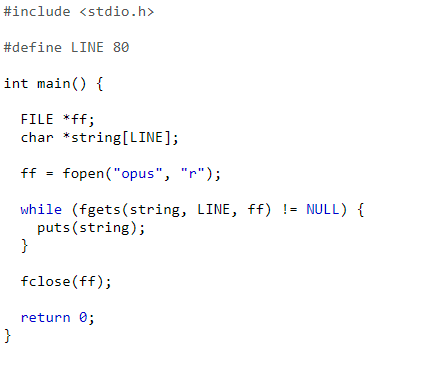
Скріншот виконання коду, приклад a)

Цей код - програма на мові C, яка намагається відкрити файл "proba" для читання. Якщо відкриття вдається, програма зчитує символи з файлу та виводить їх на екран. Якщо файл не може бути відкритий, програма виводить повідомлення про це. У будь-якому випадку, після завершення операцій файл закривається.



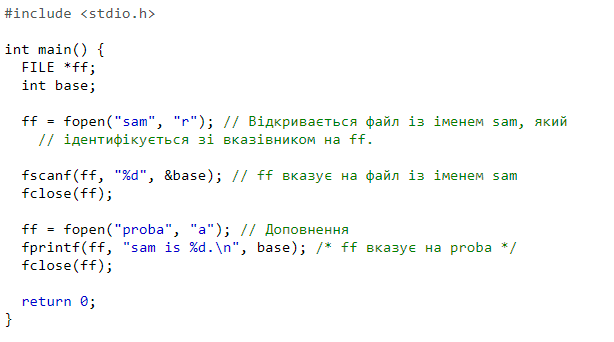
Код завдання 1, приклад б)

Ця програма відкриває файл з назвою "sam" для читання та зчитує з нього одне ціле число, яке потім записується у змінну base. Після цього вона відкриває файл "data" для дописування та записує у нього рядок, в якому зазначається число, прочитане з файлу "sam". Нарешті, обидва файли закриваються.



Код завдання 1, приклад в)

Ця програма відкриває файл "opus" для читання за допомогою функції fopen(). Потім вона використовує цикл while, щоб почергово читати рядки з файлу за допомогою функції fgets(). Кожен рядок зчитується в масив символів string розміром LINE (80 символів), і потім виводиться на екран за допомогою функції puts(). Цикл продовжується до тих пір, поки не буде досягнутий кінець файлу або якщо відбудеться помилка при читанні файлу. Наприкінці програма закриває файл за допомогою fclose().



Код завдання 1, приклад г)

Цей код відкриває файл "sam" для читання і зчитує з нього одне ціле число. Потім він закриває файл "sam". Після цього відкривається файл "proba" для дописування. Прочитане число з файлу "sam" дописується до файлу "proba" разом з певним текстом. Нарешті, обидва файли закриваються.

**Завдання 2**

Функції форматного обміну з файлами fprintf() і fscanf() є аналогами функцій printf() і scanf(), проте призначені для роботи з файлами замість стандартного введення та виведення з консолі.

Ось відмінності між ними:

fprintf() і printf():

printf(): Ця функція виводить вміст на консоль (екран).

fprintf(): Вона записує вміст у файл замість консолі. Вона має два аргументи: перший - файловий покажчик (який вказує на файл, в який буде записаний вміст), а другий - форматний рядок (такий самий, як і для printf()).

fscanf() і scanf():

scanf(): Ця функція приймає введення з консолі.

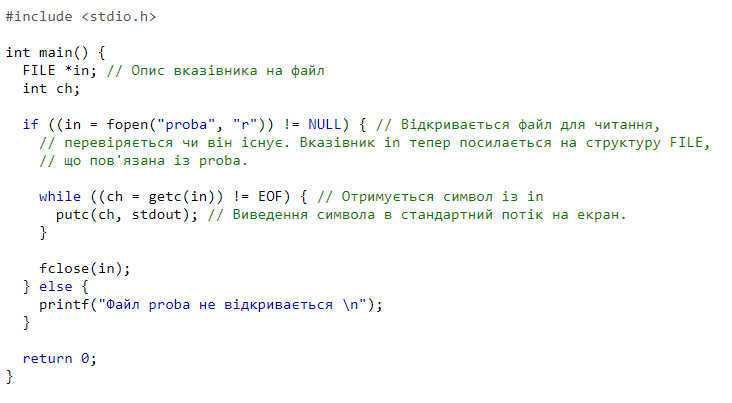
fscanf(): Вона приймає введення з файлу. Аргументи функції включають файловий покажчик (який вказує на файл, з якого буде зчитано введення) та форматний рядок (який вказує, які дані будуть зчитуватися з файлу).

Отже, головна відмінність між ними полягає у джерелі введення та місці призначення виведення: scanf() і printf() працюють з консоллю, а fscanf() і fprintf() з файлами.

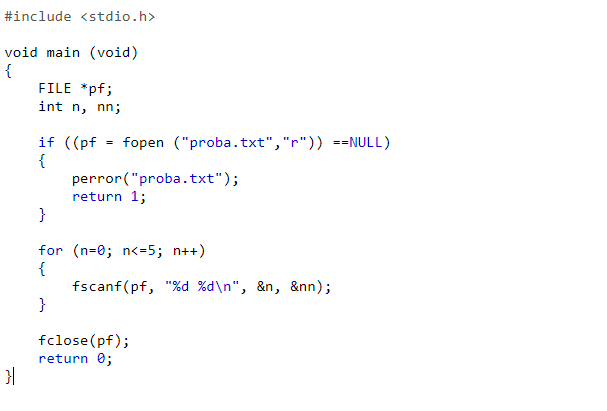
**Завдання 3**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 3, приклад 1



Код завдання 2, приклад 1

Перший код відкриває файл "proba.txt" для запису та записує числа від 0 до 5 та їхні четверті степені у цей файл. Основні етапи виконання цього коду такі:

Включення бібліотеки <stdio.h>, яка містить визначення функцій для введення та виведення даних.

Оголошення змінної типу вказівника на файл FILE \*pf; та змінної int k;.

Відкриття файлу "proba.txt" для запису за допомогою функції fopen().

Перевірка, чи вдалося відкрити файл. У разі помилки виводиться повідомлення про помилку та програма припиняє роботу.

Виконання циклу для запису чисел та їхніх четвертих степенів у файл.

Закриття файлу за допомогою функції fclose().

Другий код відкриває файл "proba.txt" для читання та зчитує дані з цього файлу. Основні етапи виконання цього коду такі:

Включення бібліотеки <stdio.h>.

Оголошення змінних типу вказівника на файл FILE \*pf; та змінних int n, nn;.

Відкриття файлу "proba.txt" для читання за допомогою функції fopen().

Перевірка, чи вдалося відкрити файл. У разі помилки виводиться повідомлення про помилку та програма припиняє роботу.

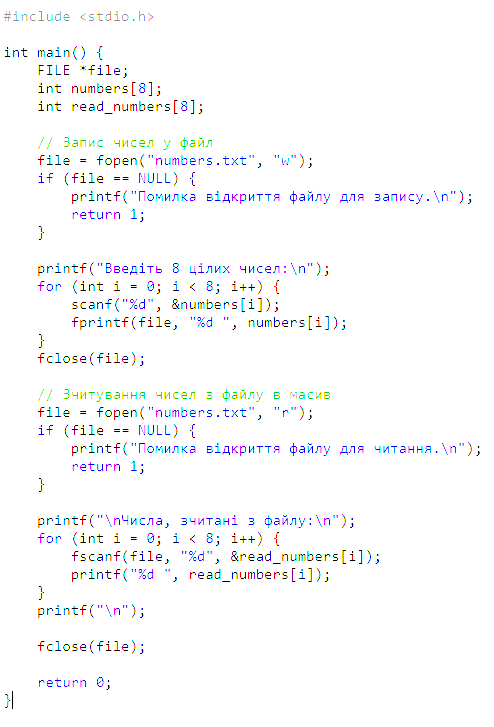
Виконання циклу для зчитування чисел та їхніх четвертих степенів з файлу. У цьому коді використовується функція fscanf().

Закриття файлу за допомогою функції fclose().

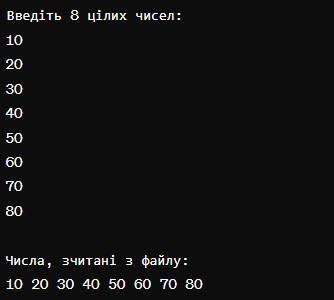
**Завдання 4**

**Реалізація:**

Програмна реалізація мовою С



Код завдання 4



Скріншот виконання коду

Цей код на мові програмування C виконує наступні дії:

Підключає необхідну бібліотеку <stdio.h>, яка містить оголошення функцій для вводу/виводу.

Оголошує головну функцію main().

Оголошує змінні:

file - вказівник на файл.

numbers[8] - масив для зберігання введених користувачем чисел.

read\_numbers[8] - масив для зберігання чисел, які будуть зчитані з файлу.

Відкриває файл "numbers.txt" для запису з використанням режиму "w". Якщо файл не відкривається, програма виводить повідомлення про помилку та завершує роботу.

Виводить повідомлення, що просить користувача ввести 8 цілих чисел.

За допомогою циклу for програма зчитує введені користувачем числа за допомогою функції scanf() і записує їх у файл за допомогою функції fprintf().

Закриває файл "numbers.txt".

Відкриває файл "numbers.txt" для читання з використанням режиму "r". Якщо файл не відкривається, програма виводить повідомлення про помилку та завершує роботу.

Виводить повідомлення про числа, які були зчитані з файлу.

За допомогою циклу for програма зчитує числа з файлу за допомогою функції fscanf() і виводить їх на екран за допомогою printf().

Закриває файл "numbers.txt".

Повертає 0, що означає успішне завершення програми.

# Контрольні питання

Суть поняття файлу в мові програмування С: У мові програмування С, файл представляє собою набір даних, який може бути збережений на зовнішньому носії (наприклад, на жорсткому диску). Він може містити будь-яку інформацію, таку як текст, зображення, аудіо або відео. У програмах на мові С файли використовуються для збереження даних, їх читання та запису.

Основні режими відкриття файлу в мові С: В мові С існує кілька режимів відкриття файлу:

"r" - відкриття для читання.

"w" - відкриття для запису (якщо файл існує, то він буде обнулений, якщо ні - створений).

"a" - відкриття для запису в кінець файлу (якщо файл існує, запис відбувається з кінця).

"r+" - відкриття для читання і запису.

"w+" - відкриття для читання і запису (якщо файл існує, то він буде обнулений, якщо ні - створений).

"a+" - відкриття для читання і запису в кінець файлу (якщо файл існує, запис відбувається з кінця).

Основні функції при роботі з файлами в мові С: У мові С для роботи з файлами використовуються наступні основні функції:

fopen() - відкриття файлу.

fclose() - закриття файлу.

fread() - читання з файлу.

fwrite() - запис в файл.

fseek() - переміщення покажчика положення в файлі.

ftell() - повертає поточну позицію покажчика положення в файлі.

rewind() - встановлює покажчик положення в файлі на початок.

Способи позиціювання в файлі в мові С: У мові С для позиціювання в файлі можна використовувати наступні методи:

Використання функції fseek(), яка дозволяє переміщати покажчик положення в файлі на вказану позицію.

Використання функції ftell(), яка повертає поточну позицію покажчика положення в файлі.

Використання функції rewind(), яка встановлює покажчик положення в файлі на початок.

Висновок:

У ході нашого дослідження різних методів потокового уведення/виведення в мові програмування C я виявив, що стандартні функції бібліотеки stdio.h, такі як scanf() та printf(), дійсно є простими та зручними для використання. Проте, вони можуть виявитися менш ефективними у випадках великого обсягу даних.

Крім того, я переконався, що робота з файловими потоками за допомогою функцій fopen(), fclose(), fread() та fwrite() дає більшу гнучкість та ефективність у керуванні потоками вводу/виводу. Ці методи можуть бути корисними при роботі з великими обсягами даних або коли потрібна більша структурованість обробки інформації.