File Handling

Q1. How to create a file in C?

- **Objective**: Understand how to create a file in C and write to it using `fopen()`.
- **Explanation**: To create a file in C, you use the `fopen()` function with the mode `"w"` (write). If the file does not exist, it will be created.

Q2. Program to create a new text file and write "Coding Age" into it.

Objective: Write a program that creates a text file and writes the string "Coding Age" to it.

Expected Output:

```
Coding Age
```

Instructions:

1. Open a file named `newfile.txt` in write mode (`"w"`).
2. Write "Coding Age" into the file.
3. Print "Coding Age" to the console after successfully writing to the file.

Q3. Program to read the contents of a text file and display them on the console.

Objective: Write a program that reads from a file and displays its content on the console.

Input:
A file named
`input.txt` with the content:

```
This is a test file.
It contains some text.
```

Expected Output:

```
This is a test file.
It contains some text.
```

Instructions:

1. Open the file `input.txt` in read mode (`"r"`).
2. Read and display the file contents to the console.

Class Work

Q1. Program to create a new text file named `output.txt` and write the string "Coding Age" into it.

Objective: Create a new file and write a string to it.

Expected Output:

```
The program should display the message "Coding Age" on the console after s
uccessfully writing to the file.
```

Instructions:

1. Create and write to the `output.txt` file.
2. Display "Coding Age" on the console after successfully writing the content.

Q2. Program to read the contents of a text file and display them on the console.

Objective: Read and display file content from `input.txt`.

Input:
A file named
`input.txt` with the content:

```
This is a test file.
It contains some text.
```

Expected Output:

```
This is a test file.
It contains some text.
```

Q3. Program to count the number of characters, words, and lines in a text file.

Objective: Create a program that counts and displays the number of characters, words, and lines in a file.

Input:
A file named
`input.txt` with the content:

```
This is a test file.
It contains some text.
```

Expected Output:

```
Characters: 38
Words: 7
Lines: 2
```

Q4. Program to append text "C is fun!" to an existing file without overwriting its content.

Objective: Learn to append content to an existing file.

Input:
A file named
`output.txt` with the content:

```
Coding Age
```

Expected Output:
After running the program, the content of
`output.txt` will be:

```
Coding Age
C is fun!
```

Home Work

Q1. Program to read a file line by line using `fgets()` and display each line.

Input:
A file named
`input.txt` with the content:

```
This is line 1.
This is line 2.
```

Expected Output:

```
This is line 1.
This is line 2.
```

Q2. Program to copy the contents of one text file to another.

Input:
A file named
`input.txt` with the content:

```
Copy this content.
```

Expected Output:
A file named
`output.txt` with the content:

```
Copy this content.
```

Q3. Program to merge the contents of two files into a third file.

Input:

- **file1.txt**: `Hello from file 1.`
- **file2.txt**: `Hello from file 2.`

Expected Output:
A file named
`merged.txt` with the content:

```
Hello from file 1.
Hello from file 2.
```

- - - Student Management System - - -

System Features

1. **Add a New Student**: Add a new student record to the file.
2. **View All Students**: Display a list of all students and their details.
3. **Search a Student**: Search for a student using their unique ID.
4. **Update Student Details**: Modify a student's information.
5. **Delete Student**: Delete a student record from the system using their ID.
6. **Mark Enrollment Status**: Change a student's enrollment status to "enrolled" or "withdrawn."

Menu Options

1. **Add Student**: Save new student records into the file.
2. **View Students**: Display all stored student records.
3. **Search Student by ID**: Find a specific student record by their unique ID.
4. **Update Student Details**: Modify an existing student’s information.
5. **Update Enrollment Status**: Mark a student as "enrolled" or "withdrawn."
6. **Exit**: Close the program.