✅ **Day : File Handling (12-8-2025)**

1. Write a program to create and write to a text file.

IPO:
Input – text from user;
Process – open file and write text;
Output – confirmation message

```
#include <stdio.h>
int main()
{
   FILE *fp;
   char text[100];
   fp = fopen("output.txt", "w");
   if (fp == NULL) { printf("Error opening file!\n"); return 1; }

   printf("Enter text to write: ");
   scanf(" %[^\n]", text);

   fprintf(fp, "%s", text);
   fclose(fp);
   printf("Data written to output.txt\n");
   return 0;
}
```

2. Write a program to read contents of a file and display.

IPO:

Input – file name;
Process – open and read characters;
Output – display file contents

```
#include <stdio.h>
int main()
{
   FILE *fp;
   char ch;
   fp = fopen("output.txt", "r");
   if (fp == NULL) {
      printf("Error: Could not open file.\n");
      return 1;
   }
   printf("File contents:\n");
   while ((ch = fgetc(fp)) != EOF)
{
      putchar(ch);
   }
   fclose(fp);
   return 0;
}
```

3. Write a program to count number of lines in a file.

IPO:
Input – file name;
Process – count newline characters;
Output – display number of lines

```
#include <stdio.h>
int main()
{
   FILE *fp;
   char ch;
   int lines = 0;
   fp = fopen("output.txt", "r");
   if (fp == NULL) { printf("Error opening file!\n"); return 1; }

   while ((ch = fgetc(fp)) != EOF)
      if (ch == '\n') lines++;
   fclose(fp);
   printf("Number of lines: %d\n", lines + 1);
   return 0;
}
```

4. Write a program to copy contents from one file to another.

Input: Read contents from `source.txt`
Process: Copy each character from source file to destination file
Output: Create `destination.txt` with the same contents

```
#include <stdio.h>
int main()
{ FILE *src, *dest; char ch;
src = fopen("source.txt", "r");
if (src == NULL) {
   printf("Error opening source file!\n");
   return 1;
}
dest = fopen("destination.txt", "w");
if (dest == NULL) {
   printf("Error opening destination file!\n");
   fclose(src);
   return 1;
}
while ((ch = fgetc(src)) != EOF) {
   fputc(ch, dest);
}
printf("File copied successfully!\n");
fclose(src);
fclose(dest);
return 0;
}
```

5. Write a program to append text to a file.

IPO:

Input: Text entered by the user

Process: Open file in append mode and write the text at the end

Output: Updated file with the new text appended

```
#include <stdio.h>
int main()
{
   FILE *fp;
   char text[200];

   fp = fopen("myfile.txt", "a");

   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }
   printf("Enter text to append: ");
   fgets(text, sizeof(text), stdin);
   fputs(text, fp);
   fclose(fp);
   printf("Text appended successfully.\n");
   return 0;
}
```

6.Write a program to count vowels in a file.

IPO (single line):
Input: File contents
Process: Check each character for vowels
Output: Total vowel count.

```
#include <stdio.h>
#include <ctype.h>
int main()
{
   FILE *fp;
   char ch;
   int count = 0;
   fp = fopen("myfile.txt", "r");
   if (fp == NULL)
{
      printf("Error opening file!\n");
      return 1;
   }
   while ((ch = fgetc(fp)) != EOF)
{
      ch = tolower(ch);
      if (ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u')
         count++;
   }
   fclose(fp);
   printf("Number of vowels in file: %d\n", count);
   return 0;
}
```

7.Write a program to read integers from a file and find the sum.

IPO (single line):
Input: Integers from file
Process: Add each integer
Output: Sum of integers.

```
#include <stdio.h>
int main()
{
   FILE *fp;
   int num, sum = 0;
   fp = fopen("numbers.txt", "r");
   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }

   while (fscanf(fp, "%d", &num) == 1)
{
      sum += num;
   }
   fclose(fp);
   printf("Sum of integers in file: %d\n", sum);
   return 0;
}
```

8.Write a program to read a structure from a file.

IPO (single line):
Input: Structure data from file
Process: Read structure into memory
Output: Display structure contents.

```
#include <stdio.h>
struct Student
{
   char name[50];
   int age;
   float marks;
};
int main()
{
   struct Student s;
   FILE *fp;
   fp = fopen("student.dat", "rb");
   if (fp == NULL)
{
      printf("Error opening file!\n");
      return 1;
   }
   fread(&s, sizeof(s), 1, fp);
   fclose(fp);
   printf("Name: %s\nAge: %d\nMarks: %.2f\n", s.name, s.age, s.marks);
   return 0;
}
```

```c
#include <stdio.h>
#include <string.h>
int main()
{
   char names[50][50], temp[50];
   int n = 0, i, j;
   FILE *fp;
   fp = fopen("names.txt", "r");
   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }
   while (fgets(names[n], sizeof(names[n]), fp))
{
      names[n][strcspn(names[n], "\n")] = 0;
      n++;
   }
   fclose(fp);
   for (i = 0; i < n - 1; i++)
{
      for (j = i + 1; j < n; j++)
{
         if (strcmp(names[i], names[j]) > 0)
{
            strcpy(temp, names[i]);
            strcpy(names[i], names[j]);
            strcpy(names[j], temp);
         }
      }
   }
   printf("\nSorted Names:\n");
   for (i = 0; i < n; i++) {
      printf("%s\n", names[i]);
   }
```

9.Write a program to sort names stored in a file.

IPO (single line):

Input: Names from file
Process: Read names, sort alphabetically
Output: Display sorted names.

10.Write a program to search for a word in a file.

IPO :
Input: Word from user & file contents
Process: Read each word, compare with input
Output: Display if found or not.

```c
#include <stdio.h>
#include <string.h>
int main()
{
   FILE *fp;
   char word[50], temp[200];
   int found = 0;
   printf("Enter word to search: ");
   scanf("%s", word);

   fp = fopen("sample.txt", "r");
   if (fp == NULL) {
      printf("Error opening file!\n");
      return 1;
   }

   while (fscanf(fp, "%s", temp) != EOF)
{
      if (strcmp(temp, word) == 0)
{
         found = 1;
         break;
      }
   }
   fclose(fp);

   if (found)
      printf("Word found in file.\n");
   else
      printf("Word not found.\n");
   return 0;
}
```