BINARY FILES

A binary file is a file that uses all 8 bits of a byte for storing the information .
It is the form which can be interpreted and understood by the computer.

The only difference between the text file and binary file is the data contain in
text file can be recognized by the word processor while binary file data can't be recognized by a word processor.

Binary File Modes

1.wb(write)
this opens a binary file in write mode.
SYNTAX:
fp=fopen("data.dat","wb");

2.rb(read)
this opens a binary file in read mode
SYNTAX:
fp=fopen("data.dat","rb");

3.ab(append)
this opens a binary file in a Append mode i.e. data can be added at the end of file.
SYNTAX:
fp=fopen("data.dat","ab");

Writing and Reading data from a Binary file

fwrite()

The fwrite() function is used to write records (sequence of bytes) to the file.
A record may be an array or a structure.

Syntax:

```
fwrite( ptr, int size, int n, FILE *fp );
```

The fwrite() function takes four arguments.

ptr : ptr is the reference of an array or a structure stored in memory.
size : size is the total number of bytes to be written.
n : n is number of times a record will be written.
FILE* : FILE* is a file where the records will be written in binary mode.

fread()

The fread() function is used to read bytes form the file.

Syntax:
fread(ptr, int size, int n, FILE *fp);

The fread() function takes four arguments.
ptr : ptr is the reference of an array or a structure where data will be stored after reading.
size : size is the total number of bytes to be read from file.
n : n is number of times a record will be read.
FILE* : FILE* is a file where the records will be read.

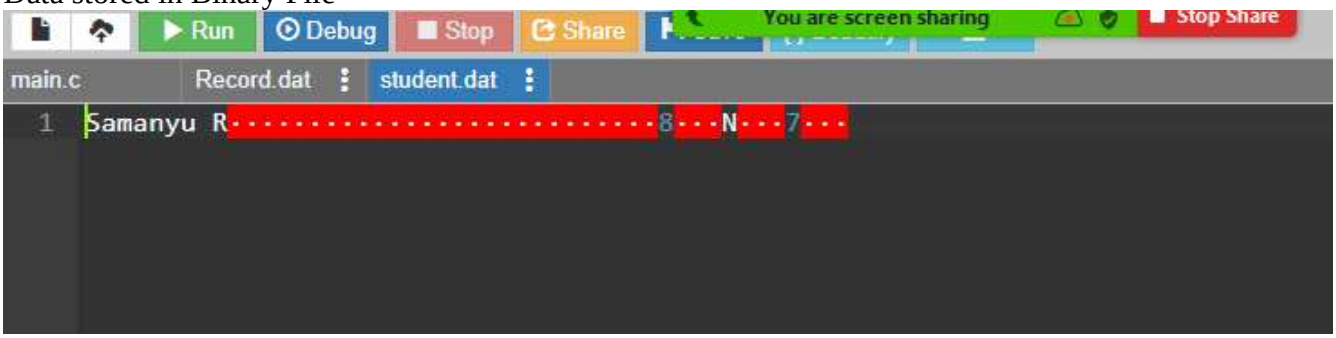
// 32 program from Lab Syllabus
// Write a C program to write the record list of Student type into a binary file student.dat.
//Re-open the file read the records from the file and display on the screen.

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

struct student
{
char name[30];
int rollno;
int m1,m2,m3;
}s1;

void main()
{
FILE *fp;
fp=fopen("student.dat","wb"); //opening binary file in writing mode

printf("Enter student detailsn");
printf("Name:");
scanf(" %[^\n]",s1.name);
printf("Roll No:");
scanf("%d",&s1.rollno);

printf("Marks in three subjects:");
scanf("%d%d%d",&s1.m1,&s1.m2,&s1.m3);

fwrite(&s1,sizeof(s1),1,fp);     //writing content on to the binary file
printf("\nRecord has been added successfully");
fclose(fp);

 fp = fopen("student.dat","rb");
 printf("\n\tName\tRno\tMarks\n");

 while(fread(&s1,sizeof(s1),1,fp)>0)
 printf("\n\t%s\t%d\t%d\t%d\t%d\t%d",s1.name,s1.rollno,s1.m1,s1.m2,s1.m3);
fclose(fp);
```

}

```
input
main.c:37:35: warning: format '%d' expects a matching 'int' argument [-Wformat=]
Enter student detailsnName:Samanyu R
Roll No:789
Marks in three subjects:56
78
55

Record has been added successfully
        Name    Rno     Marks

        Samanyu R       789     56      78      55      -620524400

...Program finished with exit code 0
Press ENTER to exit console.
```

Data stored in Binary File




```
// Demo on fwrite()
#include<stdio.h>
#include<stdlib.h>

int main()
{
   FILE *fp;
   int a;
   char name[20];
   fp = fopen("data.dat","wb");
    printf("Enter Number");
    scanf("%d",&a);
    printf("Enter Name\n");
    scanf(" %[^\n]",name);

   fwrite(&fp,sizeof(a),2,fp );
   fclose(fp);
}
```

```
// 28 Program - Write a C program to read the content of a given text file and count the number of characters, words and
//lines in it (Read the file name as command line argument).
#include <stdio.h>
#include <stdlib.h>

int main()
{
   FILE *fp;

   char ch;
   int characters, words, lines;

   fp = fopen("file1.txt", "r");

   characters = words = lines = 0;
   while ((ch = fgetc(fp)) != EOF)
   {
      characters++;

      /* Check new line */
      if (ch == '\n' || ch == '\0')
         lines++;

      /* Check words */
      if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
         words++;
   }

   /* Increment words and lines for last word */
  /* if (characters > 0)
   {
      words++;
      lines++;
   }
*/

   printf("\n");
   printf("Total characters = %d\n", characters);
   printf("Total words      = %d\n", words);
   printf("Total lines      = %d\n", lines);


   /* Close files to release resources */
   fclose(fp);
```

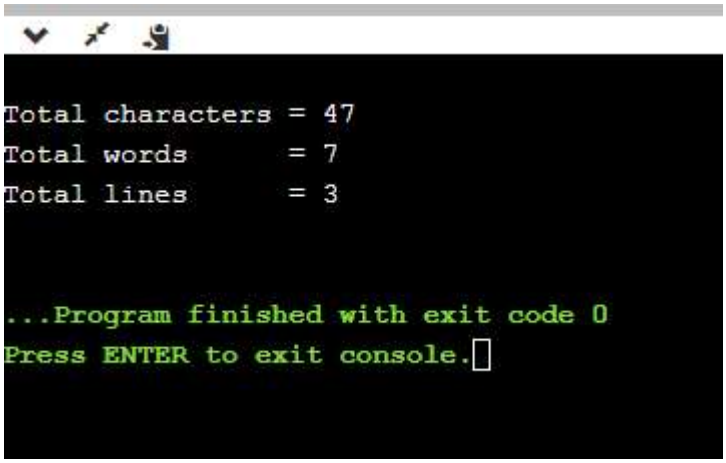
```
    return 0;
}
```

// 28 Program - Write a C program to read the content of a given text file and count the number of characters, words and
//lines in it (Read the file name as command line argument).

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

int main()
{
    FILE *fp;

    char ch;
    int characters, words, lines;

    fp = fopen("file1.txt", "r");

    characters = words = lines = 0;
    while ((ch = fgetc(fp)) != EOF)
    {
        characters++;

        /* Check new line */
        if (ch == '\n' || ch == '\0')
            lines++;

        /* Check words */
        if (ch == ' ' || ch == '\t' || ch == '\n' || ch == '\0')
            words++;
    }

    /* Increment words and lines for last word */
   /* if (characters > 0)
    {
        words++;
        lines++;
    }
*/

    printf("\n");
    printf("Total characters = %d\n", characters);
    printf("Total words      = %d\n", words);
```

```
    printf("Total lines      = %d\n", lines);


    /* Close files to release resources */
    fclose(fp);

    return 0;
}
```

output




```
// AIM - Write a C program to read the content of a given text file,
//convert all lower case letters into upper case and display it on the screen
#include <stdio.h>
#include<string.h>
int main()
{

    FILE* fptr; char ch;
    fptr = fopen("gfg.txt","r");

    // converting into upper case
    while ((ch = fgetc(fptr)) != EOF)
        {
        ch = toupper(ch);
        printf("%c", ch);
        }

    fclose(fptr);

    return 0;
}
```

text file : gfg.txt

```
Cvr College of Engineering
```

Output :
CVR COLLEGE OF ENGINEERING