

UNIT 8

Files and File Handling

The input/output function, Like `printf()`, `scanf()`, `getchar()`, `putchar()`, `gets()`, `puts()`, are known as console oriented I/O functions which always use keyboard for input device. While using these library function the entire data is lost when either the program is terminated or computer is turned off. Again it becomes cumbersome and time consuming to handle large volume of data through keyboard. It takes lot of time to enter the entire data. If user makes a mistake while entering the data he/she has to start from beginning again. If the same data is to be entered again at some later age, again we have to enter the same data. These problems invite the concept of data file in which data can be stored on disks and read whenever necessary, without destroying data.

What is file?

File is a place on the disk where a group of related data is stored. The data file allow us to store information permanently and to access and alter the information whenever necessary.

Why file handling is needed in c program?

When problem involves large volume of data and in such situations the console oriented i/o operations pose two major problems.

- ✓ It becomes cumbersome and time consuming to handle large volume of data through terminals.
- ✓ The entire data is lost when either a program is terminated or the computer is turned off.

To solve these problems the concept of data file is introduced in which data can be stored on disks and read whenever necessary, without destroying data. However, if we have a file containing all the data, we can easily access the contents of the file using few commands in C.

C supports a number of functions that have the ability to perform the basic file operations, which include:

- Naming a file
- Opening a file
- Writing data to file
- Reading data from a file and
- Closing the file

There are two distinct ways to perform file operations in C. The first one as the low-level I/O and used UNIX system calls. The second method is referred to as the high-level I/O operation and uses functions that are available in C library.

Text file vs Binary file

- I. **Text files:** Text file is a human readable sequence of characters and the words they form that can be encoded into computer-readable format such as ASCII. A text file is also known as ASCII file and can be read by any word processor. Text files store information in consecutive characters. These characters can be interpreted as individual data items or as a component of strings or numbers.
- II. **Binary files:** A binary file is made up of machine-readable symbols that is made up of 0's and 1's. The binary file must be interpreted by the program that understands in advance exactly how it is formatted. Binary files are organized into blocks containing contiguous bytes of information. These blocks represent more complex data structures, such as arrays and structures.

Text file	Binary file
1. Text file is human readable because everything is stored in terms of text.	1. In binary file everything is written in terms of 0 and 1, therefore binary file is not human readable.
2. Every text file is terminated with EOF (End of File), whose ASCII value is 26.	2. There is no such special character present in the binary mode files to mark the end of file.
3. In text format some conversions have to take place while transferring data between main memory and file. For example, while writing to text file newline('\n') has to be converted to underlying systems end of file representation and while reading from text file this end of line representation is converted to '\n'. All these conversions are done by the library functions.	3. Data is stored the same way as it is represented in memory so no conversions have to take place while transferring data between memory and file.
4. In text file, the text and characters are stored one character per byte. For example, the integer value 12345 will occupy 2 bytes in memory but it will occupy 5 bytes in text file.	4. In binary file, the integer value 12345 will occupy 2 bytes in memory as well as in file.
5. The extension for text file is .txt.	5. The extension for binary file is .dat.
6. File opening modes for text file are r, w, a, r+, w+ and a+.	6. File opening modes for binary file are rb, wb, ab, rb+, wb+ and ab+.

File Pointer

What is the significance of file pointer in file handling?

- ✓ A file pointer is a pointer to a structure, which contains information about the file, including its name, current position of the file, whether the file is being read or written, and whether errors or end of the file have occurred.
- ✓ The user does not need to know the details, because the definitions obtained from `stdio.h` include a structure declaration called `FILE`.

The only declaration needed for a file pointer is symbolized by

```
FILE *fptr;
```

This says that `fptr` is the file pointer that points to a `FILE` structure.

Opening and closing a file

- ✓ Before a program can write to a file or read from a file, the program must open it.
- ✓ Opening a file establishes a link between program and the operating system. This provides the operating system the name of a file and the mode in which file is to be opened.
- ✓ While working with high level data file, we need buffer area where information is stored temporarily in the course of transferring data between computer memory and data file.
- ✓ The process of establishing connection between the program and the file is called opening a file.
- ✓ A structure named `FILE` is defined in the file `stdio.h` that contains all the information about file like name, status, buffer size, current position and of file status etc. A file pointer is a pointer to a structure of type `FILE`.
- ✓ Whenever a file is opened, a structure of type `FILE` associated with it and the file pointer that points to this structure identifies this file.

The function **`fopen()`** is used to open the file. The buffer area is established by:

```
FILE *ptr_variable;
```

And file is opened by using the following syntax:

```
ptr_variable = fopen("file_name", "file_opening_mode");
```

Here, **`file_name`** is the name of the file we intend to open and **`file_opening_mode`** specifies the purpose of opening file.

For example:

```
FILE *fptr;  
fptr=fopen("info.txt","w+");
```

The file that was opened using **`fopen()`** function must be closed when no more operations are to be performed on it. After closing the file the connection between file and the program is broken. On closing the file, all buffers associated with it are flushed and can be available for other files.

Its syntax is:

```
fclose(ptr_variable);
```

for example:

```
fclose(fptr);
```

File opening modes

Describe the different file opening modes in C.

File opening modes specifies the way in which a file should be opened. In other words, it specifies the purpose of opening a file. Let us discuss the file opening mode for text file.

File mode	Meaning of mode	During Inexistence of file
"r"	Opens an existing file for reading purpose only	If the file does not exist, fopen() returns NULL.
"w"	Opens a file for writing purpose	If the file exists, its contents are overwritten (contents are deleted first and written). If the file does not exist, it will be created.
"a"	Opens an existing file for appending purpose (i.e. addition of new content at the end of the existing file)	If the file does not exist, it will be created.
"r+"	Opens an existing file for both reading and writing purpose.	If the file does not exist, fopen() returns NULL.
"w+"	Opens file for reading and writing purpose.	If the file exists, its contents are overwritten. If the file does not exist, it will be created.
"a+"	Opens an existing file for both reading and appending purpose.	If the file does not exist, it will be created.

Note: The file opening modes in binary files are similar to text mode. Character b is added to each mode to indicate the binary mode. For example: rb, wb, ab, rb+, wb+, ab+.

For example:

```
FILE *fptr;
```

```
fptr=fopen("sample.txt", r);
```

Here, file named sample.txt is opening for reading mode only.

Similarly,

```
FILE *fptr;
```

```
fptr=fopen("hello.dat",wb+);
```

Here, file named hello.dat is opening for both reading and writing.

Errors in Opening Files

If an error occurs in opening a file, then `fopen()` returns `NULL`.so we can check for any errors in opening by checking the return value of `fopen()`.

```
FILE *fptr;
fptr=fopen("text.dat", "wb");
if(fptr=NULL)
{
printf("Error in opening the file");
exit(1);
}
```

End of File

The file reading functions need to know the end of file so that they can stop reading. When the end of a file is reached, the operating system sends an end-of-file signal to the program. When the program receives this signal the file reading functions returns `EOF`, which is a constant defined in the file `stdio.h` and its value is `-1`. `EOF` is an integer value, so make sure that the return value of the function is assigned to an integer variable. Note that the `EOF` is not present at the end of the file, it is returned by file reading functions when end of file is reached

Structure of file program

```
int main()
{
FILE *fptr;
fptr=fopen("file_name","file_opening_mode");
if(fptr=NULL)
{
printf("Error in opening the file");
exit(1);
}
.....
.....
fclose(fptr);
return 0;
}
```

Note:

rewind()

This function is use to move the file position indicator to the beginning of the file.

1) Character I/O functions

Using character I/O functions data can be read from a file or written to a file one character at a time.

fgetc(): It is used to read a character from a file.

Syntax: **char_variable=fgetc(file_ptr_variable);**

fputc(): It is used to write a character to file.

Syntax: **fputc(char_variable,file_ptr_variable);**

Write a program to read characters from keyboard and write character to a file. (until user hits the enter key)

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char ch;
    FILE *fptr;
    fptr=fopen("sample.txt","w");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter some characters:");
    while((ch=getchar())!='\n')
    {
        fputc(ch,fptr);
    }
    fclose(fptr);
    return 0;
}
```

Write a Program to read a character stored in the file name sample.txt and display it on screen.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char ch;
    FILE *fptr;
    fptr=fopen("sample.txt","r");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
}
```

```

printf("The characters from the file are:");
while((ch=fgetc(fp))!=EOF)
{
    putchar(ch);
}
fclose(fp);
return 0;
}

```

Write a program to read characters from the keyboard, store them in a file, then read the characters from the file and display them.

```

#include<stdio.h>
#include<stdlib.h>
int main()
{
    char ch;
    FILE *fp;
    fp=fopen("sample.txt","w+");
    if(fp==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter some characters:");
    while((ch=getchar())!='\n')
    {
        fputc(ch,fp);
    }
    rewind(fp);
    printf("The characters from the file are:");
    while((ch=fgetc(fp))!=EOF)
    {
        putchar(ch);
    }
    fclose(fp);
    return 0;
}

```

2) String Input/ Output functions

Using string I/O functions, data can be read from a file or written to a file in the form of array of characters.

`fgets()`: It is used to read a string from a file

syntax: `fgets(string_variable,int_value,file_ptr_variable);`

Here, `int_value` denotes the number of characters in string .The functions read a string from a file representing `file_ptr_variable` and stores in a variable string variable.

`fputs()`:It is used to write a string to a file

syntax: `fputs(string_variable,file_ptr_variable);`

Here, contents in the string variable is written to a file representing a `file_ptr_variable`.

Write a program to create a file named “info.txt” and write “I am an Engineer”.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *fptr;
    fptr=fopen("info.txt","w");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    fputs("I am an Engineer",fptr);
    fclose(fptr);
    return 0;
}
```


Write a Program to write a string data to a file

OR

Write a program to create a new file named "test.txt" and write some string to the file.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
    FILE *fptr;
    char str[50];

    fptr=fopen("sample.txt","w");
    if(fptr==NULL)
    {
        printf("Error in opening the file\n");
        exit(1);
    }
    printf("Enter the strings\n");
    while(strlen(gets(str))>0)
    {
        fputs(str,fptr);
        fputs("\n",fptr);
    }
    fclose(fptr);
    return 0;
}
```

Write a Program to read string data from file

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    FILE *fptr;
    char str[50];
    fptr=fopen("sample.txt","r");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("The information in file are\n");
    while(fgets(str,50,fptr)!=NULL)
    {
        puts(str);
    }
    fclose(fptr);
    return 0;
}
```

Write a Program to write strings to a file and read from the file and display it.

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
int main()
{
char str[50];
FILE *fptr;
fptr=fopen("text.txt","w+");
if(fptr==NULL)
{
printf("Error in opening the file");
exit(1);
}
printf("Enter the strings\n");
while((strlen(gets(str))>0))
{
fputs(str,fptr);
fputs("\n",fptr);
}
rewind(fptr);
printf("The information in file are\n");
while(fgets(str,50,fptr)!=NULL)
{
puts(str);
}
fclose(fptr);
return 0;
}
```

3) Formatted Input /Output

These functions are used to read numbers, characters or string from a file or write them to a file in a format as per requirement.

fprintf() It is formatted output function which is used to write integer, float ,char or string data to a file.

Syntax: fprintf(file_ptr_variable,"control_string",list_variables);

fscanf() It is formatted input function which is used to read integer ,float, char or string data from a file.

Syntax: fscanf(file_ptr_variable,"format_specifier",&list_variables);

Write a program to read the name, address, and salary of n employees from the user and store them in the file named employee.txt.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char name[20],address[20];
    float salary;
    int i,n;
    FILE *fptr;
    fptr=fopen("employee.txt","w");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter the number of employees\n");
    scanf("%d",&n);
    for(i=0;i<n;i++)
    {
        printf("Enter the name,address and salary of employee %d\n",i+1);
        scanf("%s%s%f",name,address,&salary);
        fprintf(fptr,"%s\t%s\t%f\n",name,address,salary);
    }
    fclose(fptr);
    return 0;
}
```

Write a program to read the name, address, and salary of n employees from the file named employee.txt and display the information.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    char name[20],address[20];
    float salary;
    int i;
    FILE *fptr;
    fptr=fopen("employee.txt","r");
    if(fptr==NULL)
    {
        printf("Error in opening file");
        exit(1);
    }
}
```

```
printf("Name\tAddress\tsalary\n");
while(fscanf(fp, "%s%s%f", name, address, &salary) != EOF)
{
printf("%s\t%s\t%f\n", name, address, salary);
}
fclose(fp);
return 0;
}
```

Write a program to read the name, address, and salary of n employees from the user, store them in the file named employee.txt, and then read the information from the file and display it.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
char name[20],address[20];
float salary;
int i,n;
FILE *fp;
fp=fopen("employee.txt","w+");
if(fp==NULL)
{
printf("Error in opening the file");
exit(1);
}
printf("Enter the number of employees\n");
scanf("%d",&n);
for(i=0;i<n;i++)
{
printf("Enter the name,address and salary of employee %d\n",i+1);
scanf("%s%s%f",name,address,&salary);
fprintf(fp, "%s\t%s\t%f\n", name, address, salary);
}
rewind(fp);
printf("Name\tAddress\tsalary\n");
while(fscanf(fp, "%s%s%f", name, address, &salary) != EOF)
{
printf("%s\t%s\t%f\n", name, address, salary);
}
fclose(fp);
return 0;
}
```

Assignment:

- ✓ Write a program to create a file "hello.txt", write data info to the file and finally read from the file.
- ✓ Write a Program to open a new file and read name, address and telephone number of 10 employees from a user and write to a file named "employee.txt."

Write a Program to read name, age and height of person from the keyboard until user want to enter the information and stored them in the file named 'info.dat' and read information from file and display them on screen.

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
char name[20];
char choice;
int age;
float height;
FILE *fptr;
fptr=fopen("info.dat","wb+");
if(fptr==NULL)
{
printf("Error in opening the file");
exit(1);
}
do
{
printf("Enter the name");
scanf("%s",name);
printf("Enter the age");
scanf("%d",&age);
printf("Enter the height" );
scanf("%f",&height);
printf("Do you want to continue?");
fflush(stdin);
scanf("%c",&choice);
fprintf(fptr,"%s\t%d\t%f\n",name,age,height);
}while(choice=='Y' || choice=='y');
rewind(fptr);
printf("\nName\tAge\tHeight");
while(fscanf(fptr,"%s%d%f",name,&age,&height)!=EOF)
{
printf("\n%s\t%d\t%f",name,age,height);
}
fclose(fptr);
return 0;
}
```

Record I/O

- ✓ Record i/o writes numbers to files in binary format so that integers are stored in 2 bytes, floating point numbers are stored in 4 bytes and so on.
- ✓ It permits reading or writing multiple data values in the form of arrays, structures and array of structures once.
- ✓ Thus, the arrays, structures, array of structures etc. can be read from a file or written to a file as a single unit using record i/o.

There are functions `fwrite()` and `fread()` for writing and reading structure (record) in a file on a disk.

`fwrite ()` used for writing entire block to a given file.

Syntax:

```
fwrite(&ptr, size_of_array_or_structure, number_of_structure_or_array, fptr);
```

`fread()` is used to read an entire block from a given file.

Syntax:

```
fread(&ptr, size_of_array_or_structure, number_of_structure_or_array, fptr);
```

where,

- `ptr` is the address of an array or structure to be written
- `Size_of_array_or_structure` is an integer value that shows the size of structure or size of array which is being read or written.
- `number_of_structure_or_array` is an integer value that indicates number of arrays or structures to be written to file or read from file.
- `fptr` is a file pointer of a file opened in binary mode.

<i>Illustration of fwrite() function</i>	<i>Illustration of fread() function</i>
<pre> #include<stdio.h> #include<stdlib.h> struct student { int roll; char name[25]; float marks; }; int main() { FILE *fptr; char ch; struct student st; fptr=fopen("test.dat","wb"); if(fptr == NULL) { printf("Error in opening the file"); exit(1); } do { printf("Enter the Rollno\n"); scanf("%d",&st.roll); printf("Enter the Name\n"); scanf("%s",st.name); printf("Enter the Marks\n"); scanf("%f",&st.marks); fwrite(&st,sizeof(st),1,fptr); printf("Do you want to add another data (y/n):"); ch = getche(); }while(ch=='y' ch=='Y'); printf("\nData written successfully"); fclose(fptr); return 0; } </pre>	<pre> #include<stdio.h> #include<stdlib.h> struct student { int roll; char name[25]; float marks; }; int main() { FILE *fptr; char ch; struct student st; fptr = fopen("test.dat","rb"); if(fptr == NULL) { printf("Error in opening the file"); exit(1); } printf("Roll.no\tName\tMarks\n"); while(fread(&st,sizeof(st),1,fptr)==1) { printf("%d\t%s\t%f\n",st.roll,st.name,st.marks); } fclose(fptr); return 0; } </pre>

Illustration of fread/fwrite

```
#include<stdio.h>
#include<stdlib.h>
struct student
{
    int roll;
    char name[25];
    float marks;
};
int main()
{
    FILE *fptr;
    char ch;
    struct student st;
    fptr=fopen("test.dat","wb+");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    do
    {
        printf("Enter the Rollno\n");
        scanf("%d",&st.roll);
        printf("Enter the Name\n");
        scanf("%s",st.name);
        printf("Enter the Marks\n");
        scanf("%f",&st.marks);
        fwrite(&st,sizeof(st),1,fptr);
        printf("Do you want to add another data (y/n):\n");
        ch = getche();
    }while(ch=='y' || ch=='Y');
    printf("Data written successfully\n");
    rewind(fptr);
    printf("Roll\tName\tMarks\n");
    while(fread(&st,sizeof(st),1,fptr)==1)
    {
        printf("%d\t%s\t%f\n",st.roll,st.name,st.marks);
    }
    fclose(fptr);
    return 0;
}
```

MODEL 1 (WRITE TO THE FILE AND READ FROM THE FILE)

Hint:

- ✓ Read data from the user and store it in the file (using fwrite).
- ✓ Read data from the file and display it on the screen (using fread).

- 1) Create a file called "university .dat". Input n records of colleges in a structure having collegename, location, and no of faculties of Pokhara University. Now display name of colleges whose location is Kathmandu.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct university
{
    char name[20];
    char location[20];
    int nooffaculties;
};
int main()
{
    struct university u[100];
    int i,n;
    FILE *fptr;
    fptr=fopen("university.dat","wb+");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter number of colleges\n");
    scanf("%d",&n);
    printf("Enter records of %d college\n",n);
    for(i=0;i<n;i++)
    {
        printf("Enter college name\n");
        gets(u[i].name);
        printf("Enter location\n");
        gets(u[i].location);
        printf("Enter number of faculties\n");
        scanf("%d",&u[i].nooffaculties);
    }
    fwrite(&u,sizeof(u),n,fptr);
    rewind(fptr);
    printf("The name of colleges whose location is kathmandu are:\n");
    fread(&u,sizeof(u),n,fptr);
    for(i=0;i<n;i++)
    {
        if(strcmp(u[i].location,"kathmandu")==0)
        {
            printf("\n%s",u[i].name);
        }
    }
    fclose(fptr);
    return 0;
}
```

Note: In the above program, the entire structure is written at once to the file and read at once from the file. This approach is suggested and preferred when working with structures.

Alternative solution-1

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct university
{
    char name[20];
    char location[20];
    int nooffaculties;
};
int main()
{
    struct university u[100];
    int i,n;
    FILE *fptr;
    fptr=fopen("university.dat","wb+");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter number of colleges\n");
    scanf("%d",&n);
    printf("Enter records of %d college\n",n);
    for(i=0;i<n;i++)
    {
        printf("Enter college name\n");
        gets(u[i].name);
        printf("Enter location\n");
        gets(u[i].location);
        printf("Enter number of faculties\n");
        scanf("%d",&u[i].nooffaculties);
        fwrite(&u[i],sizeof(u[i]),1,fptr);
    }
    rewind(fptr);
    printf("The name of colleges whose location is kathmandu are:\n");
    for(i=0;i<n;i++)
    {
        fread(&u[i],sizeof(u[i]),1,fptr);
        if(strcmp(u[i].location,"kathmandu")==0)
        {
            printf("%s\n",u[i].name);
        }
    }
}
```

```

fclose(fptr);
return 0;
}

```

Note: In the above program writes and reads one structure at a time to and from the file. An array of structures is not necessary if the program processes one structure at a time. In such cases, you can also follow the alternative approach, Solution-2, where only a single structure variable is used in the program.

Alternative solution-2

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct university
{
    char name[20];
    char location[20];
    int nooffaculties;
};
int main()
{
    struct university u;
    int i,n;
    FILE *fptr;
    fptr=fopen("university.dat","wb+");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter number of colleges\n");
    scanf("%d",&n);
    printf("Enter records of %d college\n",n);
    for(i=0;i<n;i++)
    {
        printf("Enter college name\n");
        gets(u.name);
        printf("Enter location\n");
        gets(u.location);
        printf("Enter number of faculties\n");
        scanf("%d",&u.nooffaculties);
        fwrite(&u,sizeof(u),1,fptr);
    }
    rewind(fptr);

```

```

printf("The name of colleges whose location is kathmandu are:\n");
while(fread(&u,sizeof(u),1,fptr)==1)
{
    if(strcmp(u.location,"kathmandu")==0)
    {
        printf("%s\n",u.name);
    }
}
fclose(fptr);
return 0;
}

```

2) Write a program to create a structure, read the name, author, and price of 500 books from the user, and store the information in the file named 'library.dat.' Then, read the information from the file and print the name and price of books whose price is above 300.

```

#include<stdio.h>
#include<stdlib.h>
struct book
{
    char name[20];
    char author[20];
    float price;
};
int main()
{
    int i;
    struct book b[500];
    FILE *fptr;
    fptr=fopen("library.dat","wb+");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    for(i=0;i<500;i++)
    {
        printf("Enter the records of book %d\n",i+1);
        printf("Enter the book name\n");
        gets(b[i].name);
        printf("Enter the authors name\n");
        gets(b[i].author);
        printf("Enter the books price\n");
        scanf("%f",&b[i].price);
    }
    fwrite(&b,sizeof(b),500,fptr);
    rewind(fptr);
    fread(&b,sizeof(b),500,fptr);
}

```

```

printf("Book which price is above 300 are\n");
printf("Name\tprice\n");
for(i=0;i<500;i++)
{
    if(b[i].price>300)
    {
        printf("\n%s\t%f",b[i].name,b[i].price);
    }
}
fclose(fp);
return 0;
}

```

3) Write a program to input name, address, faculty, program and GPA(in maximum 4.0) of 500 students and store them in 'RESULT.DAT' data file and display the records of those student whose faculty is 'Engineering' and GPA>3.5.

```

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
struct student
{
    char name[20];
    char address[20];
    char faculty[20];
    char program[20];
    float gpa;
};
int main()
{
    struct student st[500];
    int i;
    FILE *fp;
    fp=fopen("result.dat","wb+");
    if(fp==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Enter records of 500 students");
    for(i=0;i<500;i++)
    {
        printf("Enter the records of student %d\n",i+1);
        printf("Enter student name\n");
        gets(st[i].name);
        printf("Enter the address\n");
        gets(st[i].address);
        printf("Enter the faculty\n");
        gets(st[i].faculty);
    }
}

```

```

printf("Enter the Program\n");
gets(st[i].program);
do
{
printf("Enter the gpa\n");
scanf("%f",&st[i].gpa);
}while(st[i].gpa<0 || st[i].gpa>4);
}
fwrite(&st,sizeof(st),500,fptr);
rewind(fptr);
printf("Information student of faculty Engineerig with GPA greater than 3.5 are:\n");
printf("Name\tAddress\tFaculty\tProgram\tGPA\n");
fread(&st,sizeof(st),500,fptr);
for(i=0;i<500;i++)
{
if(st[i].gpa>3.5&&strcmp(st[i].faculty,"Engineering")==0)
{
printf("\n%s\t%s\t%s\t%s\t%f",st[i].name,st[i].address,st[i].faculty,st[i].program,st[i].gpa);
}
}
return 0;
}

```

4) Consider the following structure:

Roll.No	Name	Address	Faculty	Date of Birth		
				mm	dd	yy

Write a program to create “student.txt” file to store the above records for 100 students and display the records of student who are not from pokhara.

```

#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct dob
{
int mm;
int dd;
int yy;
};
struct student
{
int rollno;
char name[20];
char address[20];
char faculty[20];
struct dob d;
};

```

```

int main()
{
    int i;
    struct student st[100];
    FILE *fptr;
    fptr=fopen("student.txt","wb+");
    if(fptr=NULL)
    {
        printf("Error in opening the file\n");
        exit(1);
    }
    printf("Enter the records of 100 students\n");
    for(i=0;i<100;i++)
    {
        printf("Enter records of student %d\n",i+1);
        printf("Enter the rollno\n");
        scanf("%d",&st[i].rollno);
        printf("Enter the name\n");
        gets(st[i].name);
        printf("Enter the address\n");
        gets(st[i].address);
        printf("Enter the faculty\n");
        gets(st[i].faculty);
        printf("Enter the year of birthday\n");
        scanf("%d",&st[i].d.yy);
        printf("Enter the month of birthday\n");
        scanf("%d",&st[i].d.mm);
        printf("Enter the day of birthday\n");
        scanf("%d",&st[i].d.dd);
    }
    fwrite(&st,sizeof(st),100,fptr);
    rewind(fptr);
    printf("\nThe information of students who are not from pokhara");
    printf("\nRollno\tName\tAddress\tFaculty\tDate of birth");
    fread(&st,sizeof(st),100,fptr);
    for(i=0;i<100;i++)
    {
        if(strcmp(st[i].address,"pokhara")!=0)
        {
            printf("\n%d\t%s\t%s\t%d\t%d\t%d",st[i].rollno,st[i].name,st[i].address,st[i].faculty,
            st[i].d.mm,st[i].d.dd,st[i].d.yy);
        }
    }
    fclose(fptr);
    return 0;
}

```

Assignment:

- 1) Create a structure with data members college name, location and number of faculties. Now read the information of 50 colleges affiliated to Pokhara university and write them into the file named university.dat and while retrieving ,display the information of colleges whose address is Kathmandu.
- 2) Write a Program to create a structure Teacher with attributes name, address, faculty and salary. Input records of 50 Teachers and store them in a "Teacher.txt" file and display the records of teachers whose address is "Gorkha".
- 3) Write a program to read the record of n students (name,roll no,grade and university) and store it into a file named student.txt and display the records of those student whose university is Pokhara University.
- 4) Write a program by using structure that includes using structure that includes AccNO, CName, CAddress, CTelno and balance for 100 customers of ABC financial institution and store in account.txt file and finally display the information of the customers who have balance greater than Rs.10,000 .

[Note : use long int for accno, ctelno]

- 5) Write a program to create structure for the following data for cricket game.(Country name, Player name, playing type(eg. bating ,balling or both),Number of matches played by player and salary).Save the information in a fie named "cricket.txt" and display the information of those players who had played more than 10 matches.
- 6) Write a program to create structure named football for the following data for football game(country name, player name, team name, number of matches played by player and salary).Save the information in a file named "football.txt" and display the information of those players whose country name is Portugal and had played more than 10 matches.
- 7) Create a structure named Employee with structure members name, eid, address and gender. Structure need to read information for 50 employees. Write all content into the file info.dat and while retrieving display the information of those employees whose address is "Kathmandu".
- 8) Create a structure for the following data:
ID, name, address, salary and joining date (dd/mm/yy)and Write a Program to input 100 employees information and store it in file employee.dat and now read records from the file and display records of those employee whose address is pokhara.

MODEL 2(WRITE TO THE FILE ONLY)

Hint:

- ✓ Read data from the user and store it in the file (using fwrite).

- 1) Write a program to create a structure, read the name, author, and price of 500 books from the user, and store the information in the file named 'library.dat.'

```
#include<stdio.h>
#include<stdlib.h>
struct book
{
    char name[20];
    char author[20];
    float price;
};
int main()
{
    int i;
    struct book b[500];
    FILE *fptr;
    fptr=fopen("library.dat","wb");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    for(i=0;i<500;i++)
    {
        printf("Enter the records of book %d\n",i+1);
        printf("Enter the book name\n");
        gets(b[i].name);
        printf("Enter the authors name\n");
        gets(b[i].author);
        printf("Enter the books price\n");
        scanf("%f",&b[i].price);
    }
    fwrite(&b,sizeof(b),500,fptr);
    fclose(fptr);
    return 0;
}
```

2) Write a program to create structure for the following data for student (RN, Name, phone, address and semester). Read the 10 students by user and write only those students whose semester is 1 in file "student.txt".

```
#include<stdio.h>
#include<stdlib.h>
struct student
{
    char name[20];
    int rollno;
    long int phone;
    char address[20];
    int semester;
};
int main()
{
    int i;
    struct student st[10];
    FILE *fptr;
    fptr=fopen("student.txt","wb");
    if(fptr==NULL)
    {
        printf ("Error in opening the file");
        exit(1);
    }
    printf("Enter records of 10 students\n");
    for(i=0;i<10;i++)
    {

        printf("Enter the name\n");
        gets(st[i].name);
        printf("Enter rollno\n");
        scanf("%d",&st[i].rollno);
        printf("Enter Phone number\n");
        scanf("%ld",&st[i].phone);
        printf("Enter the Address\n");
        gets(st[i].address);
        printf("Enter semester\n");
        scanf("%d",&st[i].semester);
        if(st[i].semester==1)
        {
            fwrite(&st[i],sizeof(st[i]),1,fptr);
        }
    }
    printf("Data written successfully");
    fclose(fptr);
    return 0;
}
```

3) Consider the following structure

Roll.No	Name	Address	Faculty	Date of Birth		
				mm	dd	yy

Write a program to create "student.txt" file to store the above records for 100 students.

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct dob
{
    int mm;
    int dd;
    int yy;
};
struct student
{
    int rollno;
    char name[20];
    char address[20];
    char faculty[20];
    struct dob d;
};
int main()
{
    int i;
    struct student st[100];
    FILE *fptr;
    fptr=fopen("student.txt","wb");
    if(fptr=NULL)
    {
        printf("Error in opening the file\n");
        exit(1);
    }
    printf("Enter the records of 100 students\n");
    for(i=0;i<100;i++)
    {
        printf("Enter records of student %d\n",i+1);
        printf("Enter the rollno\n");
        scanf("%d",&st[i].rollno);
        printf("Enter the name\n");
        gets(st[i].name);
        printf("Enter the address\n");
        gets(st[i].address);
        printf("Enter the faculty\n");
        gets(st[i].faculty);
```

```

printf("Enter the year of birthday\n");
scanf("%d",&st[i].d.yy);
printf("Enter the month of birthday\n");
scanf("%d",&st[i].d.mm);
printf("Enter the day of birthday\n");
scanf("%d",&st[i].d.dd);
}
fwrite(&st,sizeof(st),100,fptr);
fclose(fptr);
return 0;
}

```

Assignment:

- ✓ Create a structure called goods that stores number, price, purchase date(year, month, day) and quantity. Write a Program to store information of 100 goods in the file “good.dat” .

MODEL 3(READ FROM THE FILE ONLY)

Hint:

- ✓ Read data from the file and display it on the screen (using fread).

- 1) Write a program to create a structure and read the name, author, and price of 500 books stored in the file named ‘library.dat’ and print the name and price of books whose price is above 300.

```

#include<stdio.h>
#include<stdlib.h>
struct book
{
    char name[20];
    char author[20];
    float price;
};
int main()
{
    int i;
    struct book b[500];
    FILE *fptr;
    fptr=fopen("library.dat","rb");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
}

```

```

printf("Book which price is above 300 are");
printf("\nName\tprice");
fread(&b,sizeof(b),500,fptr);
for(i=0;i<500;i++)
{
    if(b[i].price>300)
    {
        printf("\n%s\t%f",b[i].name,b[i].price);
    }
}
fclose(fptr);
return 0;
}

```

Alternative solution

(if the exact number of books to be read from the file is not known)

```

#include<stdio.h>
#include<stdlib.h>
struct book
{
    char name[20];
    char author[20];
    float price;
};
int main()
{
    int i;
    struct book b;
    FILE *fptr;
    fptr=fopen("library.dat","rb");
    if(fptr==NULL)
    {
        printf("Error in opening the file");
        exit(1);
    }
    printf("Book which price is above 300 are");
    printf("\nName\tprice");
    while(fread(&b,sizeof(b),1,fptr)==1)
    {
        if(b.price>300)
        {
            printf("\n%s\t%f",b.name,b.price);
        }
    }
    fclose(fptr);
    return 0;
}

```

MODEL 3(APPEND TO THE FILE AND READ FROM THE FILE)

Hint:

- ✓ Read data from the user and append it to the file.
- ✓ Read data from the file.

Note that in this mode, sometimes we do not have a clear idea of how many records need to be read from the file because we are adding new records to the existing ones. Although we may know how many records we are going to add, we may not know how many records are already present in the file. Therefore, we cannot use a specific number while reading from the file. Instead, we can read data from the file until there are no more records.

- 1) Write a program to input name, address, registration no, faculty and academic year of admission in university of 'n' number of students of Pokhara University and append them in data file called 'STUDENT.DAT'. Then display the records of those students by reading the records from 'STUDENT.DAT' data file who got admission in 2016.**

```
#include<stdio.h>
#include<stdlib.h>
struct student
{
    char name[20];
    char address[20];
    long int regno;
    char faculty[20];
    int year;
};
int main()
{
    struct student st;
    int i,n;
    FILE *fptr;
    fptr=fopen("student.dat","ab+");
    if(fptr==NULL)
    {
        printf("File cannot be opened");
        exit(1);
    }
    printf("Enter number of students");
    scanf("%d",&n);
    printf("Enter records of %d students of pokhara university\n",n);
    for(i=0;i<n;i++)
    {
        printf("Enter name\n");
        gets(st.name);
        printf("Enter address\n");
        gets(st.address);
```

```

printf("Enter registration number\n");
scanf("%ld",&st.regno);
printf("Enter faculty\n");
gets(st.faculty);
printf("Enter admission year\n");
scanf("%d",&st.year);
fwrite(&st,sizeof(st),1,fptr);
}
rewind(fptr);
printf("The records of student who got admission in 2016 are:\n");
printf("Name\tAddress\tReg.no.\tFaculty\tAdmission Year\n");

while(fread(&st,sizeof(st),1,fptr)==1)
{
    if(st.year==2016)
    {
        printf("%s\t%s\t%ld\t%s\t%d\n",st.name,st.address,st.regno,st.faculty,st.year);
    }
}
fclose(fptr)
return 0;
}

```

Alternative solution:

```

#include<stdio.h>
#include<stdlib.h>
struct student
{
    char name[20];
    char address[20];
    long int regno;
    char faculty[20];
    int year;
};
int main()
{
    struct student st[100],s;
    int i,n;
    FILE *fptr;
    fptr=fopen("student.dat","ab+");

    if(fptr==NULL)
    {
        printf("File cannot be opened\n");
        exit(1);
    }
}

```

```

printf("Enter number of students");
scanf("%d",&n);
printf("Enter records of %d students of pokhara university\n",n);
for(i=0;i<n;i++)
{
    printf("Enter name\n");
    gets(st[i].name);
    printf("Enter address\n");
    gets(st[i].address);
    printf("Enter registration number\n");
    scanf("%ld",&st[i].regno);
    printf("Enter faculty\n");
    gets(st[i].faculty);
    printf("Enter admission year\n");
    scanf("%d",&st[i].year);
}
fwrite(&st,sizeof(st),n,fptr);
rewind(fptr);
printf("The records of student who got admission in 2016 are:\n");
printf("\nName\tAddress\tReg.no.\tFaculty\tAdmission Year");
while(fread(&s,sizeof(s),1,fptr)==1)
{
    if(s.year==2016)
    {
        printf("\n%s\t%s\t%ld\t%s\t%d",s.name,s.address,s.regno,s.faculty,s.year);
    }
}
fclose(fptr);
return 0;
}

```

- 2) An organization has a file named “staff.txt” that has 10 records of staffs having their id, name and salary. Using the concept of structure, you are required to write a program to ask the details of three more staff and add them in the file. Also, finally you need to display all the records and the total salary paid by the organization.**

```

#include <stdio.h>
#include <stdlib.h>
struct staff
{
    int id;
    char name[50];
    float salary;
};

```



```

int main()
{
    struct staff st;
    int i;
    float sum=0;
    FILE *fptr;
    fptr = fopen("staff.txt", "ab+");
    if(fptr==NULL)
    {
        printf("File cannot be opened\n");
        exit(1);
    }
    printf("Enter three records of staff\n");
    for(i=0;i<3;i++)
    {
        printf("Enter the records of staff %d\n",i+1);
        printf("Enter staff id\n");
        scanf("%d",&st.id);
        printf("Enter staff name\n");
        scanf("%s",&st.name);
        printf("Enter staff salary\n");
        scanf("%f",&st.salary);
        fwrite(&st,sizeof(st),1,fptr);
    }
    rewind(fptr);
    printf("All records of staff are\n");
    printf("Staff id\tName\tsalary\n");
    while(fread(&st,sizeof(st),1,fptr)==1)
    {
        printf("%d\t%s\t%f\n",st.id,st.name,st.salary);
        sum=sum+st.salary;
    }
    printf("Total salary paid by the organization=%f",sum);
    fclose(fptr);
    return 0;
}

```