**Experiment No.12**

**Aim:** FILE in c

**Theory**:

In C programming, file handling is an important concept that allows you to work with files, read data from them, write data to them, and manipulate their contents. Files are a way to store data permanently on secondary storage devices like hard disks, solid-state drives, etc. In C, file handling is done using the standard library functions provided in the <stdio.h> header.

* File Pointer: To work with files, C uses a file pointer, which is a special variable that keeps track of the current position within a file. The file pointer allows you to read from or write to a file at a specific location.
* File Modes: When opening a file, you need to specify the mode in which you want to open the file. The common file modes are:

1. “r”: Read mode. Opens an existing file for reading.
2. “w”: Write mode. Creates a new file for writing or overwrites an existing file.
3. “a”: Append mode. Opens an existing file for writing at the end or creates a new file if it doesn’t exist.
4. “r+”: Read and write mode. Opens an existing file for reading and writing.
5. “w+”: Read and write mode. Creates a new file for reading and writing or overwrites an existing file.

File Operations:

* Opening a File: To open a file, you use the fopen() function and provide the file name and mode as arguments. It returns a file pointer to the opened file.
* Closing a File: After you finish working with a file, you should close it using the fclose() function. This ensures that any pending data is written to the file and resources associated with the file are released.
* Reading from a File: To read data from a file, you use functions like fscanf(), fgets(), or fgetc(), which allow you to read data in different formats (e.g., integers, strings, characters) from the file.
* Writing to a File: To write data to a file, you use functions like fprintf(), fputs(), or fputc(), which allow you to write data in different formats to the file.
* Error Handling: It’s essential to handle errors when working with files. Most file handling functions return a value that can be used to check if the operation was successful or not. Additionally, you can use the feof() and ferror() functions to check for end-of-file or error conditions, respectively.
* File Positioning: You can manipulate the position within a file using functions like fseek() and ftell(). These functions allow you to move the file pointer to a specific location or determine the current position within the file.
* File Operations with Binary Data: In addition to text files, C also allows you to work with binary files, which contain data in a raw binary format. For binary file handling, you can use functions like fwrite() and fread() to write and read binary data.

Problem statement:

Write a C program to create a electricity bill management system. Program should facilitate the end. User to create a separte file for each customer record entered with customer name as the file name. Customer record details should be :cname,address,and units-consumed. Program must dynamically. Reserve the memory for each customer record created.Provide a menu driven o/p with following

Options for the program:

Option 1-> To enter customer details and display separately each file as o/p

Option 2-> To edit units-consumed for a customer file and also display the updated content

Option 3-> To append the mode of payment to customer file & display appended content on screen

#include <stdio.h>

#include <string.h>

typedef struct customer

{

    char cname[20];

    char address[30];

    int units;

    char mode[10];

} custo;

void input\_details();

int display(char file\_name[10]);

void update();

void append();

void main()

{

    int n, i, loop = 1, op,flag=0;

    char file\_name[10];

    FILE \*fp;

    custo c;

    do

    {

        printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

        printf("\n1.Enter details and display\n");

        printf("2.Edit uints consumed and display\n");

        printf("3.Append mode of payment and display\n");

        printf("4.quit\n");

        printf("Choose option: ");

        scanf("%d", &op);

        switch (op)

        {

        case 1:

        {

            printf("\nEnter the number of records:\n");

            scanf("%d", &n);

            //c = (customer \*)malloc(n \* sizeof(customer));

            for (i = 0; i < n; i++){

                input\_details();

            }

            flag=1;

            break;

        }

        case 2:

        {

            if(flag!=1)

                {printf("First enter details\n");break;}

            else

                update();

            break;

        }

        case 3:

        {

            if(flag!=1)

                {printf("First enter details\n");break;}

            else

                append();

            break;

        }

        case 4:

            loop = 0;

            break;

        default:

            printf("Enter correct option");

            break;

        }

        fflush(stdin);

    } while (loop);

}

void input\_details()

{

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    FILE \*fp;

    char file\_name[10];

    custo c;

    printf("\nEnter the filename to create: ");

    scanf("%s", file\_name);

    fp = fopen(file\_name, "w");

    printf("\nEnter customer name :");

    scanf("%s", c.cname);

    printf("\nEnter address :");

    scanf("%s", c.address);

    printf("\nEnter units consumed : ");

    scanf("%d", &c.units);

    fprintf(fp, "%s\n%s\n%d", c.cname, c.address, c.units);

    printf("\nContents written to file successfully.....\n");

    fclose(fp);

    display(file\_name);

}

int display(char file\_name[])

{

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    FILE \*f;

    custo c;

    f=fopen(file\_name,"r");

    if (f == NULL)

    {

        printf("\nFile name dose not exist\n");

        fclose(f);

    }

    else

    {

        fscanf(f,"%s\n%s\n%d",c.cname,c.address,&c.units);

        if(feof(f)){

            printf("Name: %s\nAddress: %s\nUnits %d",c.cname,c.address,c.units);

        }

        else{

            rewind(f);

            fscanf(f,"%s\n%s\n%d\n%s",c.cname,c.address,&c.units,c.mode);

            printf("Name: %s\nAddress: %s\nUnits: %d\nMode of payment: %s",c.cname,c.address,c.units,c.mode);

        };

    }

    fclose(f);

}

void update()

{

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    FILE \*f;

    custo c;

    char file\_name[10];

    int temp;

    printf("\nEnter file name to edit: ");

    scanf("%s",file\_name);

    f = fopen(file\_name, "r");

    if(f == NULL){

        printf("\nfile dose not exist\n");

    }

    else{

        fscanf(f,"%s\n%s\n%d",c.cname,c.address,&c.units);

        printf("Enter units to update: ");

        if(feof(f)){

            fclose(f);

            f = fopen(file\_name, "w");

            scanf("%d",&c.units);

            fprintf(f,"%s\n%s\n%d",c.cname,c.address,c.units);

        }

        else{

            rewind(f);

            fscanf(f,"%s\n%s\n%d\n%s",c.cname,c.address,&c.units,c.mode);

            fclose(f);

            f = fopen(file\_name, "w");

            scanf("%d",&c.units);

            fprintf(f,"%s\n%s\n%d\n%s",c.cname,c.address,c.units,c.mode);

        }

        printf("Contents updated to file sucessfully....\n");

    }

    fclose(f);

    display(file\_name);

}

void append()

{

    printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

    FILE \*f;

    custo c;

    char file\_name[10];

    printf("\nEnter file name to append to: ");

    scanf("%s",file\_name);

    f = fopen(file\_name,"r");

    if(f == NULL){

        printf("\nfile dose not exist\n");

    }

    else{

        fscanf(f,"%s\n%s\n%d",c.cname,c.address,&c.units);

        printf("Enter mode of payment: ");

        if(feof(f)){

            fclose(f);

            f = fopen(file\_name, "a");

            scanf("%s",c.mode);

            fprintf(f,"\n%s",c.mode);

        }

        else{

            fclose(f);

            f = fopen(file\_name, "w");

            scanf("%s",c.mode);

            fprintf(f,"%s\n%s\n%d\n%s",c.cname,c.address,c.units,c.mode);

        }

        printf("Mode of payment apppended to file sucessfully....\n");

    }

    fclose(f);

    display(file\_name);

}

output:

Enter the number of customer records:

2

Enter the customer details:

Enter the filename to create: ramesh

Enter customer name :ramesh

Enter address :house1

Enter units consumed :12

Contents written to file successfully.....

-----------------------------------

Enter the filename to create: suresh

Enter customer name :suresh

Enter address :houe2

Enter units consumed :23

Contents written to file successfully.....

-----------------------------------

Enter filename to read contents: suresh

----------------------------

customer details is as follows:

Name: suresh Address: houe2 Units: 23

------------------------------------

Enter filename to update units consumed: suresh

Enter new units: 45

Content updated to file successfully.....

------------------------------------

Enter filename to append mode of payment: suresh

Enter mode of payment: cash

mode of payment appended to file successfully.........

----------------------------

Enter y to continue or n to quit

n