AIM: Implement various operations on files to solve a given problem.

Program 1

PROBLEM STATEMENT: A publishing company holds in file detail of all the books they publish. However, in the future, they wish to maintain two distinct files (i) paperbacks and (ii) hardbacks. Write a program that reads a file containing details of both paperback and hardback books and creates two files as specified above. Assume that the first character in each input record indicates if the book is paperback(p) or hardback(h) or both(b).

PROGRAM:

“All_books.txt” Files Contents:

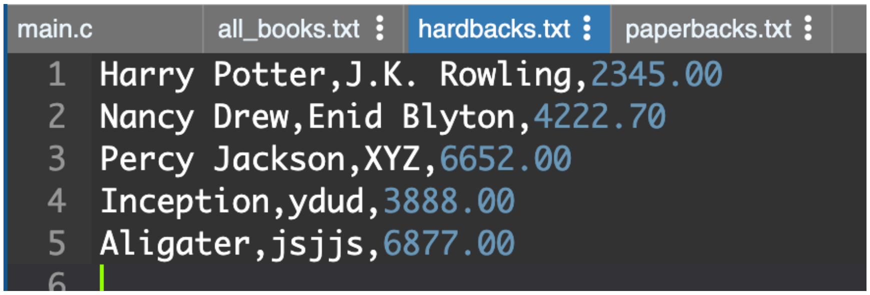
```
p,Alice in wonderland,Lewis Carroll,1234.50
h,Harry Potter,J.K. Rowling,2345
b,Nancy Drew,Enid Blyton,4222.70
h,Percy Jackson,XYZ,6652
b,Inception,ydud,3888
h,Aligater,jsjjs,6877
```

Program:

```
#include<stdio.h>
void categorise_books(char file_b[],char file_h[],char file_p[])
{
    char type,title[30],author[20];
    float price;
    FILE*fp1,*fp2,*fp3;
    fp1=fopen(file_b,"r");//Opening the file
    fp2=fopen(file_h,"w");
    fp3=fopen(file_p,"w");
    while(fscanf(fp1,"%c,%[^,],%[^,],%f\n",&type,title,author,&price)!=EOF)
    {
        if(type=='h')//To check if the book is hardback
            fprintf(fp2,"%s,%s,%.2f\n",title,author,price);
        if(type=='p') )//To check if the book is paperback
            fprintf(fp3,"%s,%s,%.2f\n",title,author,price);
        if(type=='b')
        {
            fprintf(fp2,"%s,%s,%.2f\n",title,author,price);
            fprintf(fp3,"%s,%s,%.2f\n",title,author,price);
        }
    }
    fclose(fp1);
    fclose(fp2);
    fclose(fp3);
}
int main()
{
    categorise_books("all_books.txt","hardbacks.txt","paperbacks.txt");
    return 0;
}
```

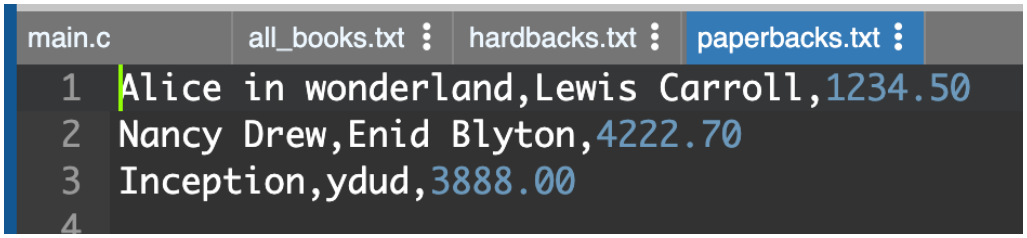
RESULT:

- **“handbacks.txt” Files Contents**



```
Harry Potter,J.K. Rowling,2345.00
Nancy Drew,Enid Blyton,4222.70
Percy Jackson,XYZ,6652.00
Inception,ydud,3888.00
Aligater,jsjjs,6877.00
```

- **“paperbacks.txt” Files Contents**



```
Alice in wonderland,Lewis Carroll,1234.50
Nancy Drew,Enid Blyton,4222.70
Inception,ydud,3888.00
```

Program 2

PROBLEM STATEMENT :

Set up a file containing vehicle records which hold registration number and owner information (name and address). Write a program which, given a vehicle’s registration number, will rapidly retrieve and print the owner information.

PROGRAM:

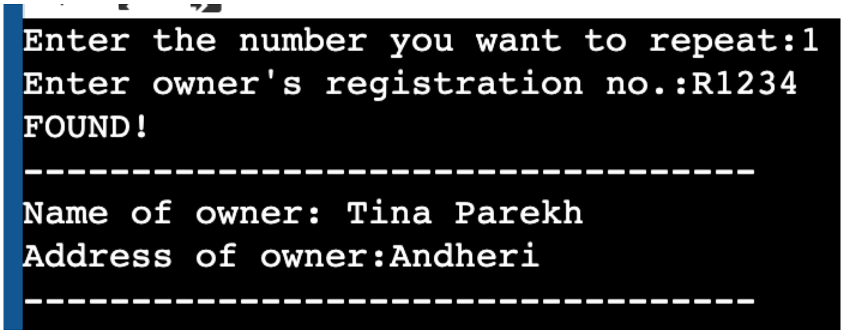
“vehicles.txt” File Contents:

```
R1234,Tina Parekh,Andheri
T1543,Falak Khan,Bandra
T2344,Rahul Roy,Khar West
R2362,Pranau Singhvi,Borivali West
R1243,RAFI Parekh,Andheri
R12354,PAKAR Parekh,Andheri
R12345,Dinchank Parekh,Andheri
R2323,Pooja Parekh,Andheri
R9999,Tanmay Parekh,Andheri
R69696,Jai Parekh,Andheri
```

Program:

```
#include<stdio.h>
#include <string.h>
struct vehicle{
        char reg_no[10],owner[30],address[40];
};
void print_owner(char filename[])
{
        FILE*fp1;
        struct vehicle v;
        int i,n,flag;
        char reg_no[10];
        printf("Enter the number you want to repeat:");
        scanf("%d",&n);
        fp1=fopen(filename,"r");
        for(i=0;i<n;i++)
        {
                flag=0;
                printf("Enter owner's registration no.:");
                scanf("%s",reg_no);
                while(fscanf(fp1,"%[^,],%[^,],%[^\n]\n",v.reg_no,v.owner,v.address)!=EOF)
                {
                        if(strcmp(v.reg_no,reg_no)==0)//Trys to find the entered register
number
                        {
                           printf("FOUND!\n----------------------------------");
                                printf("\nName of owner: %s\nAddress of owner:
%s",v.owner,v.address);
                                printf("\n----------------------------------");
                                flag=1;
                                break;
                        }
                }//end while
                if(flag==0)
                {
                        printf("\nSorry, Not found!");
                }
                rewind(fp1);
        }//end-for
        fclose(fp1);
}//end-print_owner
int main()
{
        print_owner("vehicles.txt");
        return 0;
}
```

RESULT:



```
Enter the number you want to repeat:1
Enter owner's registration no.:R1234
FOUND!
----------------------------------
Name of owner: Tina Parekh
Address of owner:Andheri
----------------------------------
```

Program 3

PROBLEM STATEMENT:

Convert All Characters into Upper Case of a File using C Program file "sample.txt" which contains "This is sample.txt file document." we will read all characters and convert them into uppercase and write into the temporary file then rename temporary file to sample.txt and remove temp.txt file.

PROGRAM:

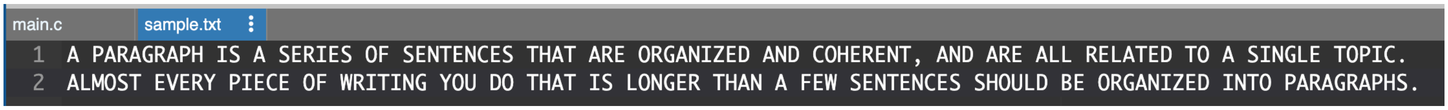
“sample.txt” File Contents:

A paragraph is a series of sentences that are organized and coherent, and are all related to a single topic. Almost every piece of writing you do that is longer than a few sentences should be organized into paragraphs.

Program:

```c
#include <stdio.h>
int main()
{
    FILE *fp1, *fp2;
    fp1 = fopen("sample.txt", "r"); // opening in read mode
    fp2 = fopen("temp.txt", "w"); // opening in write mode
    char character;
    while (fscanf(fp1, "%c", &character) != EOF)
    {
    if (character >= 97 && character <= 122)
    {
    fprintf(fp2, "%c", character - 32);
    }
    else
    {
    fprintf(fp2, "%c", character);
    }
    }
    fclose(fp1); // closing files before removing and renaming
    fclose(fp2);
    remove("sample.txt"); // deleting file
    rename("temp.txt", "sample.txt"); // renaming file
    return 0;
}
```

RESULT:



```
A PARAGRAPH IS A SERIES OF SENTENCES THAT ARE ORGANIZED AND COHERENT, AND ARE ALL RELATED TO A SINGLE TOPIC.
ALMOST EVERY PIECE OF WRITING YOU DO THAT IS LONGER THAN A FEW SENTENCES SHOULD BE ORGANIZED INTO PARAGRAPHS.
```

CONCLUSION: We learned to implement various operations on files to solve a given problem.