Name	Shubhan Singh
UID no.	2022300118
Experiment No.	10

Program 1	
PROBLEM STATEMENT :	*A publishing company holds in a file details (ISBN, Title, Author)of all the books they publish. However, in the future, they wish to maintain two distinct files (i)paperbacks (ii) hardbacks.* *Write a program which 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).*
ALGORITHM:	Algorithm for the given C program: 1. Include the necessary header files. 2. Define the main function. 3. Declare the required variables: • Integer variable 'n' to store the number of records to add. • Character variable 'type' to store the type of the book (h for hardback, p for paperback, and b for both). • Character arrays 'isbn', 'title', and 'author' to store the ISBN, title, and author of the book, respectively. • FILE pointers 'library', 'hardback', and 'paperback' to open and write to the respective files. 4. Open the 'all_books.txt', 'hardbacks.txt', and 'paperbacks.txt' files using the 'fopen' function in append mode, and assign them to 'library', 'hardback', and 'paperback' pointers, respectively. 5. Prompt the user to enter the number of records to add, and read the input using the 'scanf' function. 6. If the number of records is greater than zero, display a message asking the user to enter the records in the specified format. 7. Write the header for each of the files using the 'fprintf' function. 8. For each record to be added, do the following: • Clear the input buffer using a 'while' loop with the 'getchar' function. • Read the book type, ISBN, title, and author using the 'scanf' function.

	• Write the record to the 'all_books.txt' file using the 'fprintf' function. • If the book type is 'p' or 'b', write the record to the 'paperbacks.txt' file using the 'fprintf' function. • If the book type is 'h' or 'b', write the record to the 'hardbacks.txt' file using the 'fprintf' function. 9. Close all the files using the 'fclose' function. 10. End the main function with a 'return' statement.
PROGRAM:	

```c
#include<stdio.h>

int main(){
    int n;
    char type;
    char isbn[14],title[50],author[50];
    FILE* library,* hardback,* paperback;
    library=fopen("all_books.txt","a");
    hardback=fopen("hardbacks.txt","a");
    paperback=fopen("paperbacks.txt","a");
    printf("Enter number of records you want to add: ");
    scanf("%d",&n);
    if(n>0){printf("Enter all records in the format Type(p,h or
b),ISBN,Title,author, each in a new line\n");}
    fprintf(library,"TYPE AVAILABLE | ISBN          |
TITLE                          | AUTHOR\n");
    fprintf(hardback,"ISBN          | TITLE                          |
AUTHOR\n");
    fprintf(paperback,"ISBN          | TITLE                          |
AUTHOR\n");
    for(int i=0;i<n;i++){
        while((getchar())!='\n');
        scanf("%c",&type);
        while((getchar())!='\n');
        scanf("%[^\n]",isbn);
        while((getchar())!='\n');
        scanf("%[^\n]",title);
        while((getchar())!='\n');
        scanf("%[^\n]",author);
        fprintf(library,"%-14c | %13s | %-30s |
%s\n",type,isbn,title,author);
        if(type=='p' || type=='b'){
            fprintf(paperback,"%13s | %-30s | %s\n",isbn,title,author);
        }
        if(type=='h' || type=='b'){
            fprintf(hardback,"%13s | %-30s | %s\n",isbn,title,author);
```

```
        }
    }
    fclose(library);
    fclose(paperback);
    fclose(hardback);
    return 0;
}
```

RESULT:

```
PS C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\C programs>  & 'c:\Users\shubh\.vscode\extens
ions\ms-vscode.cpptools-1.14.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ytq0ona3.lco' '--st
dout=Microsoft-MIEngine-Out-2co33ie3.igv' '--stderr=Microsoft-MIEngine-Error-mwaflij5.hpc' '--pid=Microsoft-MIEngine-Pid-wo25xotr.upi'
'--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter number of records you want to add: 5
Enter all records in the format Type(p,h or b),ISBN,Title,author, each in a new line
p
1234567890123
To Kill a mockingbird
Harper Lee
b
1243232343443
The Trial
Franz Kafka
h
4324324243432
Nineteen Eighty Four
George Orwell
p
4324434324343
Lolita
Vladimir Nabokov
h
4332434324343
Tinker Tailor
John Le'Carre
PS C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\C programs>
```

all_books.txt U

fileio > all_books.txt

TYPE AVAILABLE	ISBN	TITLE	AUTHOR
p	1234567890123	To Kill a mockingbird	Harper Lee
b	1243232343443	The Trial	Franz Kafka
h	4324324243432	Nineteen Eighty Four	George Orwell
p	4324434324343	Lolita	Vladimir Nabokov
h	4332434324343	Tinker Tailor	John Le'Carre

paperbacks.txt U

fileio > paperbacks.txt

```
1  ISBN          | TITLE                    | AUTHOR
2  1234567890123 | To Kill a mockingbird    | Harper Lee
3  1243232343443 | The Trial                | Franz Kafka
4  4324434324343 | Lolita                   | Vladimir Nabokov
5
```

hardbacks.txt U

fileio > hardbacks.txt

```
1  ISBN          | TITLE                    | AUTHOR
2  1243232343443 | The Trial                | Franz Kafka
3  4324324243432 | Nineteen Eighty Four     | George Orwell
4  4332434324343 | Tinker Tailor            | John Le'Carre
5
```

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.*
ALGORITHM:	Algorithm for the program: 1. Define a struct called **vrec** that has three fields: **reg_no** (string), **name** (string), and **address** (string). 2. Declare the variables **x** (integer) and **n** (integer). 3. Declare a variable of type **vrec** called **temprecord**. 4. Declare a file pointer called **records**. 5. Create an infinite loop that allows the user to either append records, access records, or exit the program. Inside the loop, display the prompt "Enter 1 to append vehicle records, 0 to access them or -1 to exit: ". 6. Read the user's choice into the variable **x**. 7. If **x** is equal to -1, exit the loop. 8. If **x** is equal to 1, open the file "Vehicle_records.txt" in append mode with read and write access using the **fopen()** function and store the file pointer in the **records** variable. 9. Display the prompt "Enter number of records to add: ". 10. Read the number of records into the variable **n**. 11. Use a **for** loop to iterate **n** times. Inside the loop, display the prompt "Enter registration number, owner's name, and address (each in a new line) for record [i+1]:

	". 12. Use a **while** loop to consume any remaining characters in the input buffer. 13. Read the registration number, owner's name, and address from the user into the **temprecord** variable using the **scanf()** function. 14. Write the registration number, owner's name, and address to the file using the **fwrite()** and **fprintf()** functions. 15. Close the file using the **fclose()** function. 16. If **x** is equal to 0, open the file "Vehicle_records.txt" in append mode with read and write access using the **fopen()** function and store the file pointer in the **records** variable. 17. Declare three character pointers called **tempstr**, **tempstr2**, and **tempstr3**, and allocate memory for each using the **calloc()** function. 18. Declare a character array called **reg** of size 15. 19. Declare an integer variable called **flag** and initialize it to 0. 20. Display the prompt "Enter a registration number: ". 21. Use a **while** loop to consume any remaining characters in the input buffer. 22. Read the registration number from the user into the **reg** array using the **scanf()** function. 23. Use a **while** loop to read each record from the file. 24. Read the registration number from the file into the **tempstr** variable using the **fscanf()** function. 25. If the end of the file has been reached, exit the loop. 26. If the registration number in the **tempstr** variable is equal to the registration number entered by the user, set the **flag** variable to 1 and read the owner's name and address from the file into the **tempstr2** and **tempstr3** variables using the **fscanf()** function. 27. Display the owner's name and address to the user. 28. If the **flag** variable is still 0, display the message "Record not found!". 29. Free the memory allocated for **tempstr**, **tempstr2**, and **tempstr3** using the **free()** function. 30. Close the file using the **fclose()** function. 31. End the loop.
PROGRAM:	#include<stdio.h> #include<string.h> #include<stdlib.h> typedef struct vrecords{ char reg_no[15]; char name[50]; char address[100];

```
}vrec;
int main(){
   int x,n;
   vrec temprecord;
   FILE *records;
   records=fopen("Vehicle_records.txt","a+");
   while(1){
   printf("Enter 1 to append vehicle records, 0 to access them or -1 to exit: ");
   scanf("%d",&x);
   if(x==-1){break;}
   if(x==1){
      printf("Enter number of records to add: ");
      scanf("%d",&n);
      for(int i=0;i<n;i++){
         if(x==1){
         printf("Enter registration number, owners name and address(each in a new line) for
record %d\n",i+1);
         while((getchar())!='\n');

scanf("%s%*c%[^\n]%*c%[^\n]",temprecord.reg_no,temprecord.name,temprecord.address);
         fwrite(temprecord.reg_no,strlen(temprecord.reg_no)*sizeof(char),1,records);
         fprintf(records,"\n");
         fwrite(temprecord.name,strlen(temprecord.name)*sizeof(char),1,records);
         fprintf(records,"\n");
         fwrite(temprecord.address,strlen(temprecord.address)*sizeof(char),1,records);
         fprintf(records,"\n");
         }
      }
   }
   if(x==0){
      char *tempstr,*tempstr2,*tempstr3;
      char reg[15];
      tempstr=(char*)calloc(51,sizeof(char));
      tempstr2=(char*)calloc(51,sizeof(char));
      tempstr3=(char*)calloc(51,sizeof(char));
      int flag=0;
      printf("Enter a registration number: ");
      while((getchar())!='\n');
      scanf("%s%*c",reg);
      while(1){
```

```
            fscanf(records,"%[^\n]%*c",tempstr);
            if(feof(records)){break;}
            if(strcmp(tempstr,reg)==0){
                flag=1;
                fscanf(records,"%[^\n]%*c",tempstr2);
                printf("The name of the driver is: %s\n",tempstr2);
                fscanf(records,"%[^\n]%*c",tempstr3);
                printf("His address is: %s\n",tempstr3);
                break;
            }
        }

        if(flag==0){
            printf("Record not found!\n");
        }
        flag=0;
        free(tempstr);
        free(tempstr2);
        free(tempstr3);
    }
    }

    fclose(records);
return 0;
}
```

RESULT:

```
Enter 1 to append vehicle records, 0 to access them or -1 to exit: 1
Enter number of records to add: 3
Enter registration number, owners name and address(each in a new line) for record 1
12345678
Shubhan Singh
104A
Enter registration number, owners name and address(each in a new line) for record 2
12321232
Vikas
chakala
Enter registration number, owners name and address(each in a new line) for record 3
12547645
Nakshatra
Jogeshwari
Enter 1 to append vehicle records, 0 to access them or -1 to exit: 0
Enter a registration number: 12345678
The name of the driver is: Shubhan Singh
His address is: 104A
Enter 1 to append vehicle records, 0 to access them or -1 to exit: 0
Enter a registration number: 12547645
The name of the driver is: Nakshatra
His address is: Jogeshwari
Enter 1 to append vehicle records, 0 to access them or -1 to exit: 0
Enter a registration number: 34234233
Record not found!
Enter 1 to append vehicle records, 0 to access them or -1 to exit: -1
PS C:\Users\shubh\OneDrive - Bharatiya Vidya Bhavans Sardar Patel Institute Of Technology\C programs>
```

C vehicles_fileio.c M | *Vehicle_records.txt U* ×

fileio > Vehicle_records.txt

```
1  12345678
2  Shubhan Singh
3  104A
4  12321232
5  Vikas
6  chakala
7  12547645
8  Nakshatra
9  Jogeshwari
10
```