**Project Name : Library Research Project Application**

1. **Implementation Code**

**(Main function)**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct lib{

char name[20] ;

int id ;

int num ;}book ,a ,q;

void insert(struct lib a ,struct lib array[]);

void Delete (struct lib araay[] , int i);

void linerSearch (struct lib array[],int i );

void binarySearch(struct lib array[] , int i,struct lib q );

void sort(struct lib array[],int i,struct lib q);

void unSorted (struct lib array[] , int i );

int main()

{

struct lib array[10];

printf("Welcome To My Library\n ");

int i=0 ;

FILE \*fptr = fopen("lib.txt" ,"r");

if (fptr==NULL )

printf("couldn't open file");

else

{

fscanf(fptr,"%[^\_]%\*c",array[i].name);

fscanf(fptr,"%d",&array[i].id );

fscanf(fptr,"%d",&array[i].num);

}

while(!feof(fptr))

{

i++;

fscanf(fptr,"%[^\_]%\*c",array[i].name);

fscanf(fptr,"%d",&array[i].id );

fscanf(fptr,"%d",&array[i].num);

}

printf("1=Add a new book\n");

printf(" 2=Delete book \n");

printf(" 3=Search book by id \n");

printf(" 4=Search book by name \n");

printf(" 5=Display all books sorted by name \n");

printf(" 6=Display all books unsorted \n");

printf(" choose an operation\n ");

int c ;

scanf("%d" ,&c);

if(c==1)

insert(a,array);

else if(c==2)

Delete(array , i);

else if(c==3)

linerSearch (array,i);

else if (c==4)

binarySearch(array ,i,q );

else if(c==5)

sort(array,i,q);

else

unSorted( array , i);

int p ;

{

printf("\n do you want to make another operation ?\n");

printf("choose '1' to make operation and '-1' to end the program\n ");

scanf("%d",&p);

if (p==1)

{

system("cls");

main();

}

else

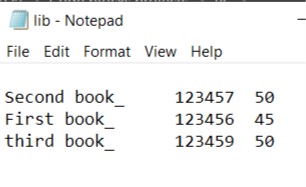
{return 0;}

}

}

1. **Function Codes and their corresponding Screenshots of Output Screen**
   1. **Create a text file**

Screenshot of the text file including some books with their ids, name, and quantity)

****

* 1. **Insert a book**

The function and the screenshot of output screen (Insert your id, complete

name, any grade as a book), text file after insertion.

void insert(struct lib a ,struct lib array[] )

{

FILE \*aptr= fopen("lib.txt" ,"a");

if (aptr==NULL)

printf("couldn't open the file ");

else

printf("please enter the name.id and the quantity of your book \n ");

getchar();

gets(a.name);

scanf("%d%d",&a.id,&a.num);

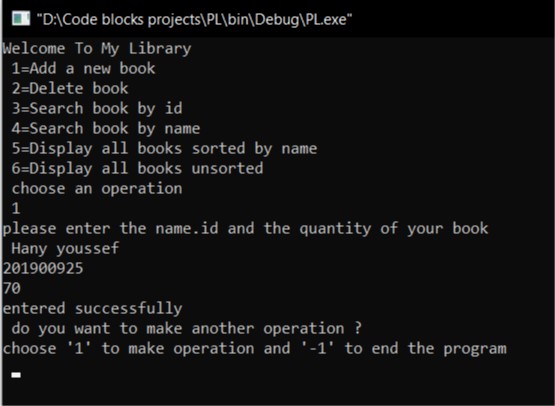
fprintf(aptr ,"\n%s\_ \t%d \t%d",a.name,a.id,a.num);

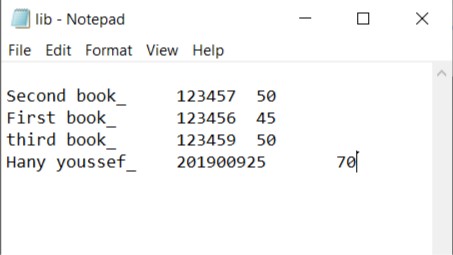
fclose(aptr);

printf("entered successfully" );

return ;

}





* 1. **Delete a book by id**

The function and the screenshot of output screen (choose an id to delete, but not your id), text file after deletion.

void Delete (struct lib array[] , int i )

{ int w=0 ,q ;

printf(" please enter the ID for the book you want to delete it\n ");

scanf("%d",&q );

w=0; while(array[w].id !=q){ w++ ;}

if (array[w].id != q)

{printf(" error 404\n");}

else

{

while(w<=i)

{

array[w]=array[w+1] ; w++ ;

}

i-- ;

}

FILE\*fptr = fopen("lib.txt","w");

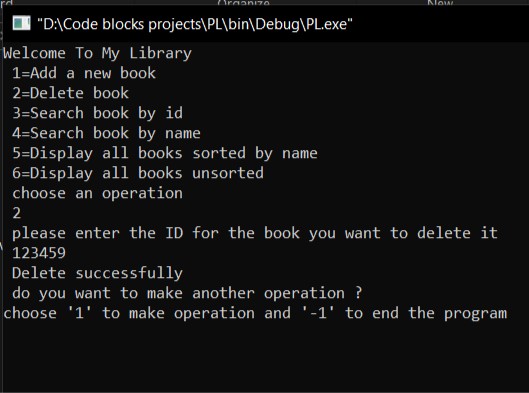
for(int v=0 ;v<=i ;v++)

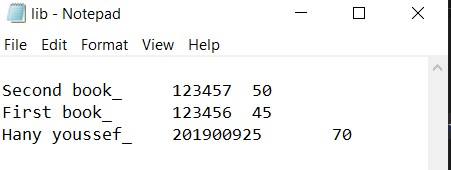
fprintf(fptr ,"%s\_\t %d\t %d\n " ,array[v].name ,array[v].id,array[v].num);

fclose(fptr);

printf(" Delete successfully");

}





* 1. **Search a book by id and display its name and quantity using linear search recursively. If not exist, display “Not found”.**

The function and the screenshot of output screen

void linerSearch(struct lib array[] ,int i )

{ int w ;

printf("enter the id to search on the book");

scanf("%d",&w);

int n=0 ;

for(n=0 ;n<=i;n++ )

{

if (array[n].id==w)

{ printf("%s\t %d\t %d\n",array[n].name ,array[n].id ,array[n].num );

return ;

}

}

if (array[n].id !=w)

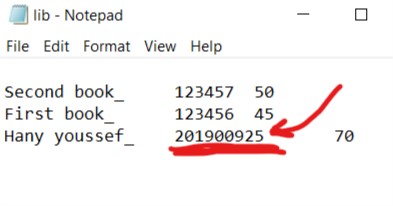
{

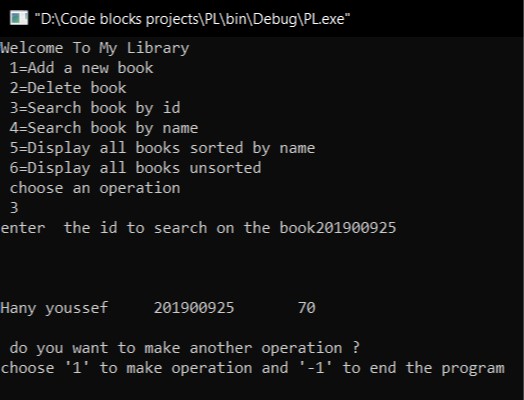
printf("Not Found ");

}

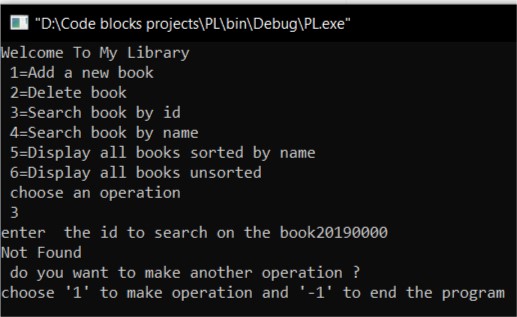
}

* Choose ***your id*** to display.





* Choose an ***id that doesn’t exist*** in your file.



* 1. **Search a book by name and display its id and quantity using binary search. If not exist, display “Not found”.**

The function and the screenshot of output screen

void binarySearch(struct lib array[] , int i,struct lib q )

{ int y ;

for (int pass=1 ; pass<=i ; pass ++)

{ for(y=0 ; y<=i-1 ; y++)

{

if (strcmp(array[y].name,array[y+1].name)>0)

{

char s[10];

strcpy(s,array[y].name);

strcpy(array[y].name , array[y+1].name);

strcpy(array[y+1].name,s);

int temp ;

temp=array[y].id ;

array[y].id = array[y+1].id ;

array[y+1].id=temp ;

temp=array[y].num ;

array[y].num = array[y+1].num ;

array[y+1].num=temp ;

}

}

}

char key[10] ;

int hight=i , low=0 ,middel ;

printf(" please enter the name of the book put ' \_ ' in the end of the name \n");

scanf("%[^\_]%\*c" , key) ;

while(low <=hight)

{

middel=(low + hight)/2 ;

strcmp(key,array[middel].name);

if (strcmp(key,array[middel].name)==0)

{printf("%s\t %d\t %d\n",array[middel].name , array[middel].id , array[middel].num);

return ;}

else if (strcmp(key,array[middel].name)==-1 )

{hight=middel-1 ;}

else

{low = middel+1 ;}

}

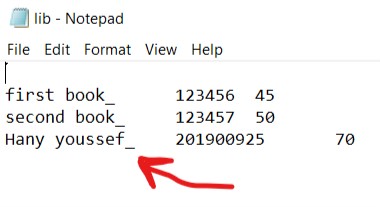
if (array[middel].name!= key)

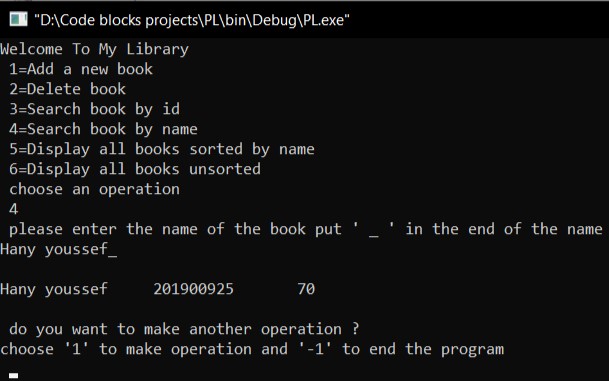
printf("Not found");

return ;

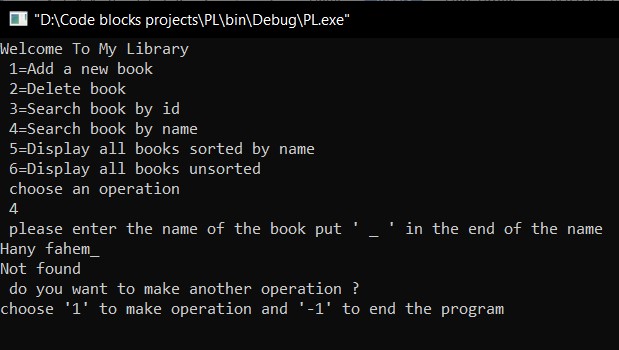
}

* Choose an ***id*** ***that exist*** to display.





* Choose an ***id that doesn’t exist*** in your file.



* 1. **Display all books sorted by name, and their corresponding ids and quantity.**

The function and the screenshot of output screen of all sorted books including your name.

void sort(struct lib array[],int i,struct lib q)

{ int y ;

for (int pass=1 ; pass<=i ; pass ++)

{

{ for(y=0 ; y<=i-1 ; y++)

if (strcmp(array[y].name,array[y+1].name)>0)

{ q=array[y];

array[y]=array[y+1] ;

array[y+1]=q ;

}

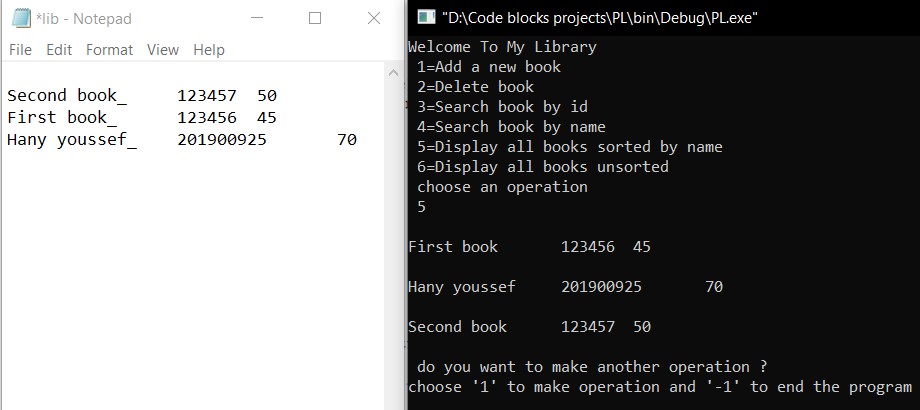
}

}

for(int x=0 ; x<=i ; x++)

printf("%s\t %d\t %d\n",array[x].name ,array[x].id ,array[x].num );

}



* 1. **Display all books unsorted, their ids, names and quantity (as entered)**

The function and the screenshot of output screen of all unsorted books including your name.

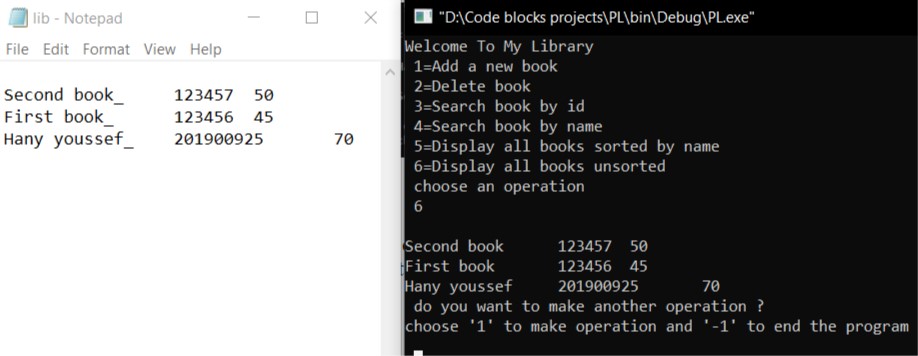
void unSorted( struct lib array[] , int i )

{ int x ;

for(int x=0 ; x<=i ; x++)

printf("%s\t %d\t %d",array[x].name ,array[x].id ,array[x].num );

}

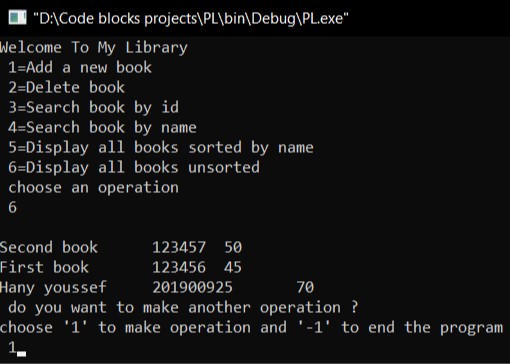
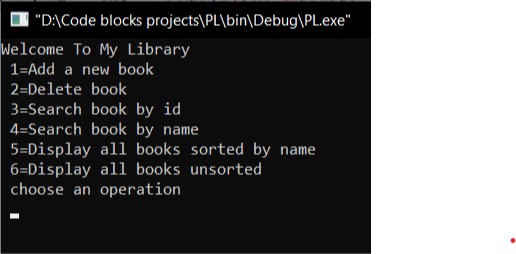
****

* 1. **Ask if you want another operation**

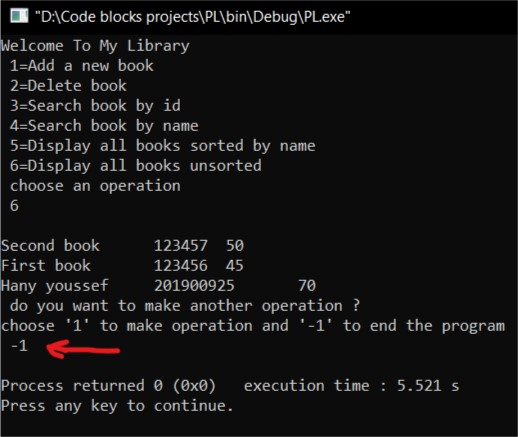
The screenshot of output screen when you ask the user if he wants another operation, reply one time by ‘yes’ and another time by ‘no’.

chose

**If choose 1 :**

** **

**If choose -1 :**

****