Name: SUNDEEP A	**SRN: PES1UG20CS445**	**Section: O**
	Date: 8-7-21	**Week Number: 10**

1 Implement Binary Search using call back when there is more than one constraint to check for.

a) Search for a number if the number is even

b) Search for a number if the number is less than 22.

Input:

enter the element to be searched

18

Output:

It is even and found at 2 position

It is less than 22 and found at 2 position

Input:

enter the element to be searched

56

Output:

It is even and found at 8 position

not found

Input:

enter the element to be searched

53

Output:

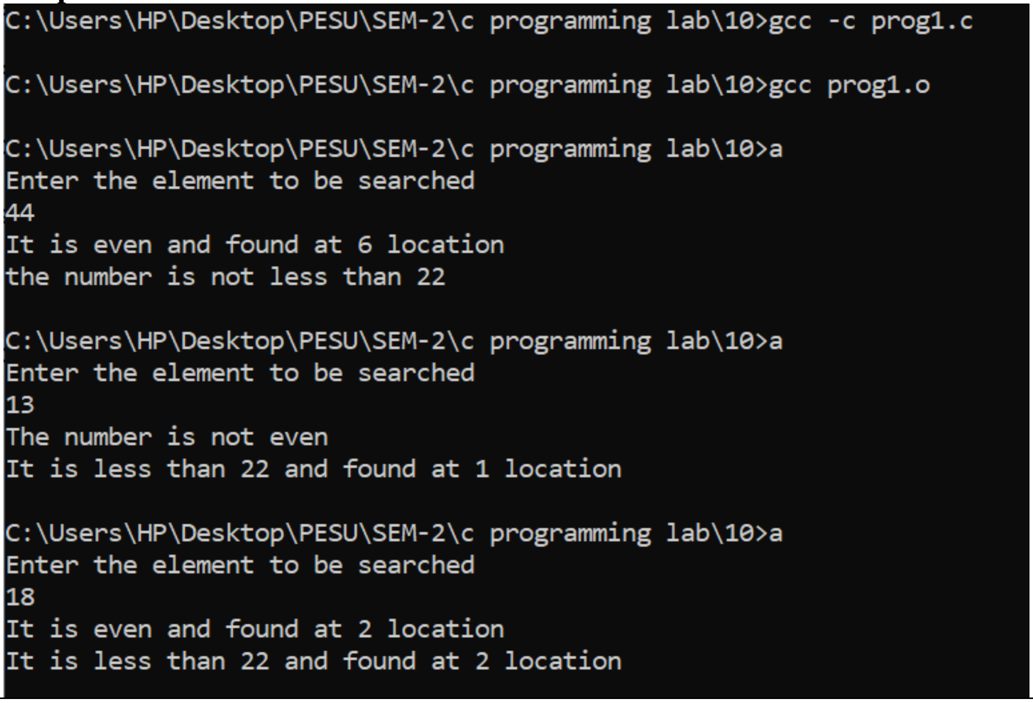
```
not found

not found
```

Program:

```
#include<stdio.h>
int my_search(int[],int,int,int,int (*p)(int));
int islessthan22(int);
int iseven(int);
int main()
{
   int a[]={11,13,18,19,22,33,44,53,56,101};
   int n=sizeof(a)/sizeof(a[0]);                    //to find the number of elements
   int key;                                         //search term

   printf("Enter the element to be searched\n");
   scanf("%d",&key);
   int pos=my_search(a,0,n-1,key,iseven);           //pos contain the return value of
my_search
   if(pos!=-1)
      printf("It is even and found at %d location\n",pos);
   else
      printf("The number is not even\n");

   pos=my_search(a,0,n-1,key,islessthan22);         //pos contain the return value of
my_search
   if(pos!=-1)
      printf("It is less than 22 and found at %d location\n",pos);
   else
      printf("the number is not less than 22\n");
   return 0;
}
int iseven(int x)
{
   int res=-1;
   if (x%2==0)
         res=1;                   //return 1 if the search element is even
   return res;
}
int islessthan22(int x)
```

```
{
   int res=-1;
   if (x<22)
         res=1;                    //return 1 if the search element is less than 22
   return res;
}
int my_search(int a[],int low,int high,int key,int (*p)(int))
{
   int mid;                        //recursive sol

   if(high>low && p(key)>0) //executes only of p(key) is true
   {
      mid=(low+high)/2;
      if(a[mid]==key)
         return mid;               //returns the position of the search element
      else if(a[mid]>key)
         return my_search(a,low,mid-1,key,p);
      else(a[mid]<key);
         return my_search(a,mid+1,high,key,p);
   }
}
```

Output Screenshot:



```
C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>gcc -c prog1.c

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>gcc prog1.o

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter the element to be searched
44
It is even and found at 6 location
the number is not less than 22

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter the element to be searched
13
The number is not even
It is less than 22 and found at 1 location

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter the element to be searched
18
It is even and found at 2 location
It is less than 22 and found at 2 location
```

2 Write a program to copy the contents of one file to another using command line arguments

(Instruction to be given in the command line)

>a abc.txt def.txt

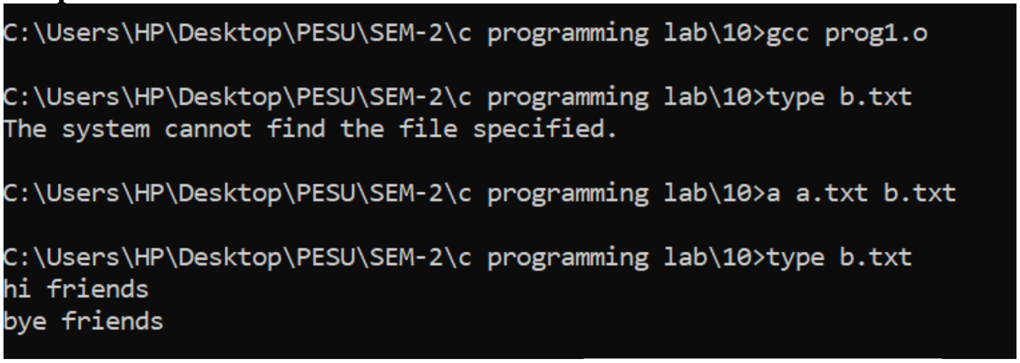
(abc.txt is the file having contents which will be copied to the file def.txt)

Program:

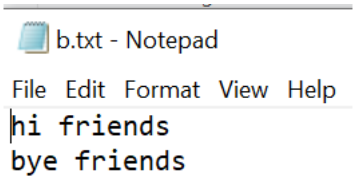
```
#include<stdio.h>
int main(int argc,char *argv[])
{
   FILE *fs,*ft;
   int ch;
   if(argc!=3)  //argument count failure
   {
      printf("Invalid number of arguments\n");
      return 1;
   }
   fs=fopen(argv[1],"r");
   if(fs==NULL)  //there is some error in opening the source file
   {
      printf("Can't file the source file\n");
      return 1;
   }
   ft=fopen(argv[2],"w");
   if(ft==NULL)  //there is some error in opening the target file
   {
      printf("Can't open the target file\n");
      fclose(fs);
      return 1;
   }
   while(1)      //copy file contents
   {
      ch=fgetc(fs);
      if(feof(fs))
         break;
      fputc(ch,ft);
   }
   printf("file copied successfully");
   fclose(fs);
```

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

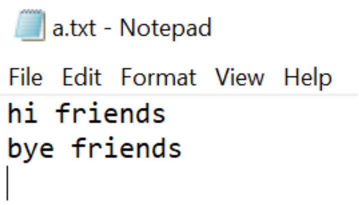
Output Screenshot:



```
C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>gcc prog1.o

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>type b.txt
The system cannot find the file specified.

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a a.txt b.txt

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>type b.txt
hi friends
bye friends
```



```
hi friends
bye friends
```



```
hi friends
bye friends
```

3 Write a program using enumerated types which when given today's date will print out tomorrow's date.

Input:

Enter a date (number 3 letter lower case month e.g. 31 jan)

30 nov

Output:

Tomorrow is 1 dec

Input:

Enter a date (number 3 letter lower case month e.g. 31 jan)

31 dec

Output:

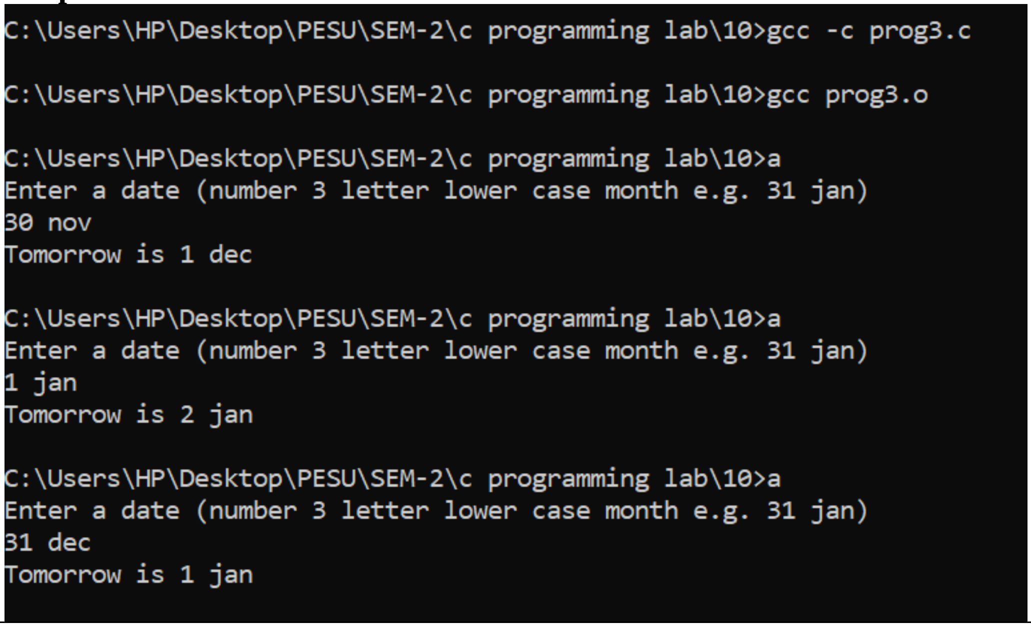
```
Tomorrow is 1 jan
```

Program:

```
#include <stdio.h>
#include<string.h>
#include<stdlib.h>

enum months {NOT_MONTH,jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
}month;                     //month names
char *month_out[] = {"NOT_MONTH","jan", "feb", "mar", "apr", "may", "jun", "jul", "aug",
"sep", "oct", "nov", "dec"};

int days_in_month[] = {-1,31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

enum months translate(char *);
void check(int, enum months);
void tomorrow(int, enum months);

int main()
{
int day;
char mon[4];
printf("Enter a date (number 3 letter lower case month e.g. 31 jan) \n");
scanf("%d %s",&day, mon);
month = translate(mon);
check(day, month);
tomorrow(day, month);
return(0);
}

enum months translate(char *m)  //finds the month entered by the user
{
if ( strcmp(m, "jan") == 0 )
 return jan;
 else
 if ( strcmp(m, "feb") == 0 )
 return feb;
 else
 if ( strcmp(m, "mar") == 0 )
 return mar;
 else
 if ( strcmp(m, "apr") == 0 )
 return apr;
 else
 if ( strcmp(m, "may") == 0 )
 return may;
 else
 if ( strcmp(m, "jun") == 0 )
 return jun;
 else
 if ( strcmp(m, "jul") == 0 )
 return jul;
 else
 if ( strcmp(m, "aug") == 0 )
 return aug;
 else
 if ( strcmp(m, "sep") == 0 )
 return sep;
 else
 if ( strcmp(m, "oct") == 0 )
 return oct;
 else
 if ( strcmp(m, "nov") == 0 )
 return nov;
 else
 if ( strcmp(m, "dec") == 0 )
 return dec;
 else
 return NOT_MONTH;
}

void check(int day, enum months month_in)  //checks if the date is valid
{
 if ( (month_in == NOT_MONTH) || (day < 1) || (day > days_in_month[month]) )
 {
printf("Error: Invalid Input %d %s\n", day, month_out[month_in]);
 exit(-1);
 }
}

void tomorrow(int day, enum months month_in)      //prints the next day's date

{
if (day < days_in_month[month_in])
```

```
        printf("Tomorrow is %d %s\n", day+1, month_out[month_in]);

else if((day == days_in_month[month_in]) && (month_out[month_in]!="dec"))
        printf("Tomorrow is 1 %s\n", month_out[month_in+1]);
else
        printf("Tomorrow is 1 %s\n","jan");
}
```

Output Screenshot:



```
C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>gcc -c prog3.c

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>gcc prog3.o

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter a date (number 3 letter lower case month e.g. 31 jan)
30 nov
Tomorrow is 1 dec

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter a date (number 3 letter lower case month e.g. 31 jan)
1 jan
Tomorrow is 2 jan

C:\Users\HP\Desktop\PESU\SEM-2\c programming lab\10>a
Enter a date (number 3 letter lower case month e.g. 31 jan)
31 dec
Tomorrow is 1 jan
```