Name: Renita Kurian	SRN: PES1UG20CS331	Section: N
	Date: 12/07/21	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.

Program:

P1.c > main()

```c
//Binary search using callback
#include <stdio.h>

int is_less_than_22(int key) // callback condition for less than 22
{
    int result = 0;
    if (key < 22) result = 1;
    return result;
}

int is_even(int key) // callback condition for even
{
    int result = 0;
    if ((key % 2) == 0) result = 1;
    return result;
}

int my_search(int* a, int l, int h, int key, int (*p)(int))// binary search
{
    int pos = -1;
    if (p(key) == 0) return pos;
    while (l <= h && pos == -1)
    {
        int mid = (l + h) / 2;
        if (a[mid] == key)
            pos = mid;
        else if (key < a[mid])
            h = mid - 1;
        else
            l = mid + 1;
    }
    return pos;
}

```

```c
int main()
{
    int a[] = {10,14,16,24,26,34,38,39,41,45,47,54,58,60,64,68,71,75,79,81,88,90,93,95,100}; //array
    int n = sizeof(a) / sizeof(int); // size of array
    int key; int result;
    printf("Enter the search element : ");
    scanf("%d", &key);
    result = my_search(a, 0, n - 1, key, is_even);
    if (result == -1)
        printf("Not found\n");
    else
        printf("Search element is even and available at position %d\n", result);
    result = my_search(a, 0, n - 1, key, is_less_than_22);
    if (result == -1)
        printf("Not found\n");
    else
        printf("Search element is less than 22 & available at position %d\n", result);
    return 0;
}

```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>gcc P1.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>a
Enter the search element : 14
Search element is even and available at position 1
Search element is less than 22 & available at position 1

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>a
Enter the search element : 64
Search element is even and available at position 14
Not found

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>a
Enter the search element : 21
Not found
Not found
```

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

P2.c > main(int, char * [])

```c
//Copy one text file into another using command line arguments
#include<stdio.h>
#include<stdlib.h>

int main(int argc, char* argv[])
{
    FILE *f1, *f2;
    int ch;
    if (argc != 3)
    {
        printf("Invalid number of arguments.");
        return 0;
    }
    f1 = fopen(argv[1], "r");
    if (f1 == NULL)
    {
        printf("Source file cannot be found");
        return 0;
    }
    f2 = fopen(argv[2], "w");
    if (f2 == NULL)
    {
        printf("target file cannot be opened");
        fclose(f1);
        return 0;
    }
    while (1)
    {
        ch = fgetc(f1);
        if (feof(f1)) break;
        fputc(ch, f2);
    }
    printf("File copy complete");
    fclose(f1);
    fclose(f2);
    return 0;
}
```

T1.txt

```
T1.txt
1    Hello World
2
```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>gcc P2.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>a T1.txt T2.txt
File copy complete
```

T2.txt

```
T2.txt
1    Hello World
2
```

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

Program:

```c
//Find the next date (using enum)
#include <stdio.h>
#include <string.h>

typedef enum month
{jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec} mn;

char months[12][4] = { "jan", "feb", "mar", "apr", "may", "jun", "jul", "aug","sep", "oct","nov", "dec" };

void findNextDate(int day, int month)
{
    if ((day == 31) && (month == 11))
        printf(" 1 jan \n");
    else if ((day == 31) && ((month == 0) || (month == 2) || (month == 4) || (month ==6) ||(month == 7) || (month == 9)))
        printf(" 1 %s \n", months[month + 1]);
    else if ((day == 30) && ((month == 3) || (month == 5) || (month == 8) || (month ==10)))
        printf(" 1 %s \n", months[month + 1]);
    else if ((day == 28) && (month == 1))
        printf(" 1 %s \n", months[month + 1]);
    else
        printf(" %d %s \n", day+1, months[month]);
}
```

```c
int main()
{
    int day; int i;
    char month[4];
    printf("Input a date (number <space> 3 letter lower case month e.g. 31 jan) : ");
    scanf(" %d %s", &day, month);
    printf("The next day is : ");
    for (i = jan; i <= dec; i++) if (strcmp(month, months[i]) == 0)
        findNextDate(day, i);
    return 0;
}
```

Output Screenshot:

```
C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>gcc P3.c

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>a
Input a date (number <space> 3 letter lower case month e.g. 31 jan) : 10 jun
The next day is :  11 jun

C:\Users\Renita Kurian\Documents\Academic\C Lab\W10>
```