

Week 10

ANIRUDH ROWJEE PES2UG20CS050

Task 1

Implement Binary Search on an array using Callbacks and check for the following constraints:

1. Search for a number that's even
2. Search for a number that's less than 22

 **main.c**

```
#include <stdio.h>
#include <stdbool.h>
#include "utils.h"

bool is_even(int t)
{
    return t % 2 == 0;
}

bool less_than_22(int t)
{
    return t < 22;
}

int main()
{
    int array_length, target;
    printf("Enter the length of the array >> ");
    scanf("%d", &array_length);

    int numbers[array_length];

    for (int i = 0; i < array_length; i++)
    {
        printf("Enter the number in position %d >> ", i+1);
        scanf("%d", &numbers[i]);
    }

    printf("\nEnter the Number to look for >> ");
    scanf("%d", &target);

    // prerequisite for binary search : sorting
    sort(numbers, array_length);

    /*
    for (int i = 0; i < array_length; i++)
        printf("%d\n", numbers[i]);
    */
}
```

```

    */

    printf("Checking if number is present and even.. \n");
    binarysearch(
        numbers, array_length, target, 0, array_length-1, &is_even
    );
    printf("Checking if number is present an less than 22.. \n");
    binarysearch(
        numbers, array_length, target, 0, array_length-1, &less_than_22
    );

    return 0;
}

```

utils.h

```

// define common result enum
typedef enum
{
    FOUND_AND_SATISFIED,
    FOUND,
    NOT_FOUND
} bs_result;

void binarysearch(
    int* numbers,
    int list_length,
    int target,
    int start,
    int end,
    bool (*test)(int test_arg)
);

// insertion sort
void sort(int* numbers, int list_length);

```

utils.c

```

#include <stdio.h>
#include <stdbool.h>
#include "utils.h"

void binarysearch(
    int* numbers,
    int list_length,
    int target,
    int start,
    int end,
    bool (*test)(int test_arg)
)
{
    /*
     * Recursive implementation of Binary Search -

```

```

    * find the number AND check if it fits the constraint
    * passed as *test
    */

// return case
if (end < start)
{
    printf("Not found.\n");
    return;
}

int middle_index = (start + end) / 2;

if (numbers[middle_index] == target)
{
    // regular binary search : return true
    if (test(numbers[middle_index]))
    {
        printf("Found at %d satisfying constraint.\n", middle_index);
    }
    else
    {
        printf("Found at %d.\n", middle_index);
    }
}
else
{
    // check if the number is higher or lower
    if (numbers[middle_index] > target)
    {
        binarysearch(
            numbers,
            list_length,
            target,
            start,
            middle_index - 1,
            test
        );
    }
    else
    {
        // lesser than case, no else-if due to logic
        binarysearch(
            numbers,
            list_length,
            target,
            middle_index + 1,
            end,
            test
        );
    }
}
}
}

```

```

void sort(int* numbers, int list_length)
{
    // insertion sort
    for (int i = 0; i < list_length; i++)

```

```

{
    for (int j = 0; j < i; j++)
    {
        if (numbers[i] < numbers[j])
        {
            // insert the number at j
            int temp = numbers[i];
            // shift each number one to the right from i - 1 to j
            for (int t = i; t > j; t--)
            {
                numbers[t] = numbers[t-1];
            }
            // insert
            numbers[j] = temp;
        }
    }
}
}
}

```

Console

```

nidavellir :: UE20CS152/week8/t1 » ./main
Enter the length of the array >> 5
Enter the number in position 1 >> 10
Enter the number in position 2 >> 32
Enter the number in position 3 >> 13
Enter the number in position 4 >> 43
Enter the number in position 5 >> 22

Enter the Number to look for >> 13
Checking if number is present and even..
Found at 1.
Checking if number is present an less than 22..
Found at 1 satisfying constraint.
nidavellir :: UE20CS152/week8/t1 » ./main
Enter the length of the array >> 6
Enter the number in position 1 >> 5
Enter the number in position 2 >> 10
Enter the number in position 3 >> 32
Enter the number in position 4 >> 13
Enter the number in position 5 >> 43
Enter the number in position 6 >> 21

Enter the Number to look for >> 21
Checking if number is present and even..
Found at 3.
Checking if number is present an less than 22..
Found at 3 satisfying constraint.
nidavellir :: UE20CS152/week8/t1 »

```

Task 2

Write a program to copy the contents of one file to another using command-line arguments. Usage:

```
$ ./main <filename1> <filename2>
```

main.c

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

int main(int argc, char* argv[])
{
    if (argc != 3)
    {
        printf("Invalid number of arguments! Only 2 allowed.\n");
    }
    else
    {
        char* source_file = argv[1];
        char* dest_file = argv[2];
        printf("Copying from %s to %s\n", source_file, dest_file);

        FILE* source = fopen(source_file, "r");
        FILE* destination = fopen(dest_file, "w");

        char c;
        while ((c = fgetc(source)) != '\0' && c != EOF)
        {
            fputc(c, destination);
        }
        printf("Successfully copied.\n");

        fclose(source);
        fclose(destination);
    }
    return 0;
}
```

Console

```
nidavellir :: UE20CS152/week8/t2 » cat input.txt
Hello, world! This is Anirudh Rowjee in Sample Text.
```

This is the second line.

```
nidavellir :: UE20CS152/week8/t2 » cat sample_out.txt
```

```
cat: sample_out.txt: No such file or directory
```

```
nidavellir :: UE20CS152/week8/t2 » ./main input.txt
```

```
Invalid number of arguments! Only 2 allowed.
```

```
nidavellir :: UE20CS152/week8/t2 » ./main input.txt sample_out.txt
```

```
Copying from input.txt to sample_out.txt
```

```
Successfully copied.
```

```
nidavellir :: UE20CS152/week8/t2 » cat sample_out.txt
```

```
Hello, world! This is Anirudh Rowjee in Sample Text.
```

This is the second line.

```
nidavellir :: UE20CS152/week8/t2 »
```

Task 3

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

main.c

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

typedef enum
{
    JAN = 31,
    FEB = 29,
    MAR = 31,
    APR = 30,
    MAY = 31,
    JUN = 30,
    JUL = 31,
    AUG = 31,
    SEP = 30,
    OCT = 31,
    NOV = 30,
    DEC = 31
} month_day;

const month_day enum_list[12] = {
    JAN,
    FEB,
    MAR,
    APR,
    MAY,
    JUN,
    JUL,
    AUG,
    SEP,
    OCT,
    NOV,
    DEC,
};

const char* month_names[12] = {
    "JAN", "FEB", "MAR", "APR", "MAY", "JUN", "JUL", "AUG", "SEP",
    "OCT", "NOV", "DEC"
};

int main()
{
    char month[15];
    int date;

    printf("Enter the Current date (DD MMM)(first 3 letters) >> ");
    scanf("%d %s", &date, month);

    int month_index;
    // get the month enum by the name
    for (int i = 0; i < 12; i++)
```

```

{
    if (strcmp(month, month_names[i]) == 0)
    {
        month_index = i;
        break;
    }
}

// printf("The current month has %d days.\n", enum_list[month_index]);
// printf("The current month has %d days.\n", (int)enum_list[month_index]);

month_day current = enum_list[month_index];

// change month if we're on the last date
if (date == (int)current)
{
    date = 1;
    month_index = (month_index + 1) % 12;
    printf("Next Date >> %d %s\n", date, month_names[month_index]);
}
else
{
    date++;
    printf("Next Date >> %d %s\n", date, month_names[month_index]);
}

return 0;
}

```

Console

```

nidavellir :: UE20CS152/week8/t3 » ./main
Enter the Current date (DD MMM)(first 3 letters) >> 27 MAR
Next Date >> 28 MAR
nidavellir :: UE20CS152/week8/t3 » ./main
Enter the Current date (DD MMM)(first 3 letters) >> 31 DEC
Next Date >> 1 JAN
nidavellir :: UE20CS152/week8/t3 » ./main
Enter the Current date (DD MMM)(first 3 letters) >> 08 MAY
Next Date >> 9 MAY
nidavellir :: UE20CS152/week8/t3 » ./main
Enter the Current date (DD MMM)(first 3 letters) >> 27 FEB
Next Date >> 28 FEB
nidavellir :: UE20CS152/week8/t3 » ./main
Enter the Current date (DD MMM)(first 3 letters) >> 01 JAN
Next Date >> 2 JAN
nidavellir :: UE20CS152/week8/t3 »

```