

DSA LAB

Week 2

Udeet Mittal
CSE C3
Roll Number 64

SOLVED EXERCISE:

1) Write a C program to implement binary search using recursion.

Filename: "binary_search_function.h"

```
int bin_search(int low,int high,int item,int a[])
{
    int mid;
    if(low>high)
        return(-1);
    else
    {
        mid=(low+high)/2;
        if(item==a[mid])
            return(mid);
        else if(item<a[mid])
            return(bin_search(low,mid-1,item,a));
        else
            return(bin_search(mid+1,high,item,a));
    }
}
```

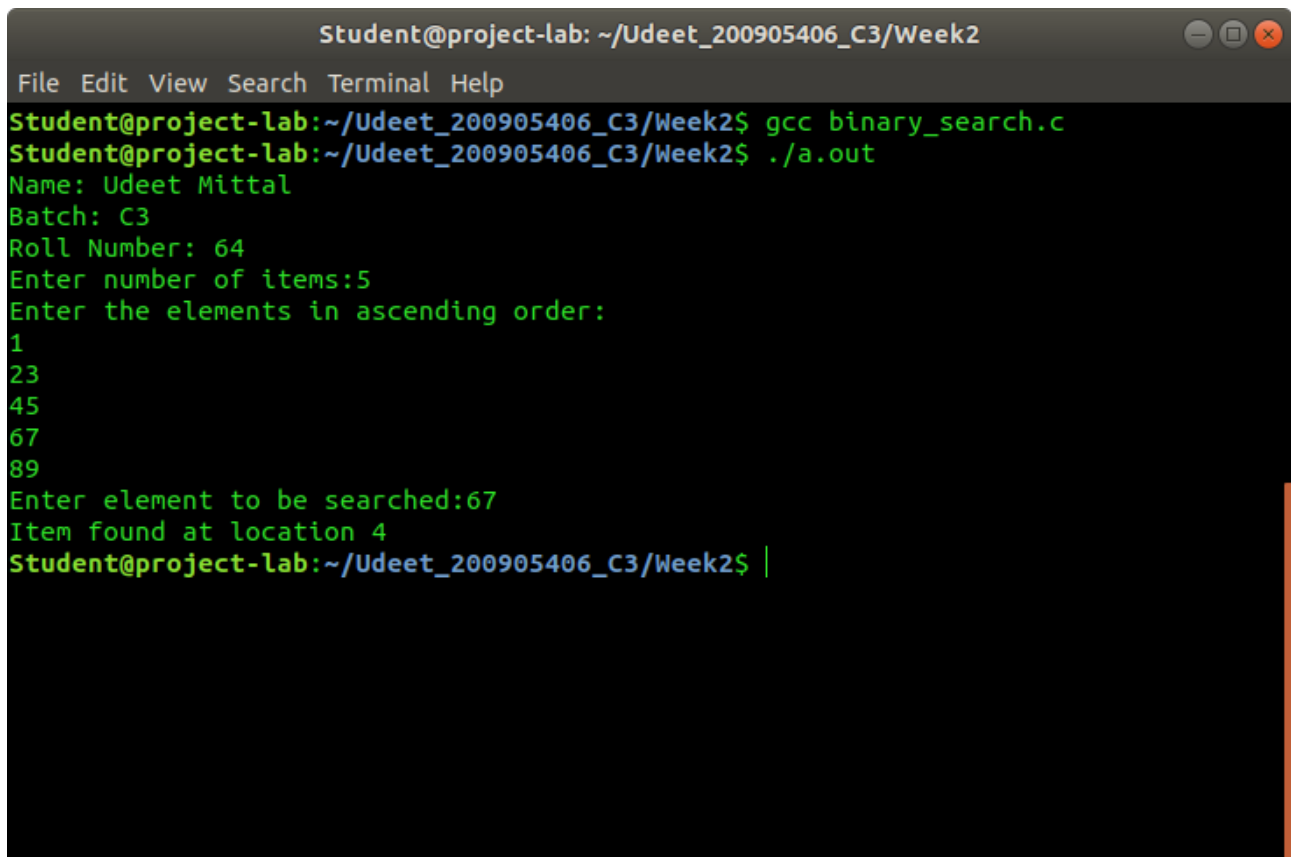
Filename: "binary_search.c"

```
#include<stdio.h>
#include "binary_search_function.h"
void main()
{
    int i, pos,a[30],n,item;
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    printf("Enter number of items:");
    scanf("%d",&n);
    printf("Enter the elements in ascending order:\n");

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

```
scanf("%d",&a[i]);

printf("Enter element to be searched:");
scanf("%d",&item);
pos=bin_search(0,n-1,item,a);
if(pos!=-1)
    printf("Item found at location %d\n",pos+1);
else
    printf("Item not found\n");
}
```

A terminal window titled "Student@project-lab: ~/Udeet_200905406_C3/Week2" with standard window controls. The terminal shows the compilation and execution of a C program. The user enters their name, batch, and roll number, then the number of items (5) and the elements in ascending order (1, 23, 45, 67, 89). They then enter the element to be searched (67), and the program outputs "Item found at location 4".

```
Student@project-lab: ~/Udeet_200905406_C3/Week2
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week2$ gcc binary_search.c
Student@project-lab:~/Udeet_200905406_C3/Week2$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter number of items:5
Enter the elements in ascending order:
1
23
45
67
89
Enter element to be searched:67
Item found at location 4
Student@project-lab:~/Udeet_200905406_C3/Week2$ |
```

LAB2 Questions:

1. Create a structure STUDENT consisting of variables of structures:

- i. DOB {day, month (use pointer), year},
- ii. STU_INFO {reg_no, name(use pointer), address},
- iii. COLLEGE {college_name (use pointer), university_name}

where structure types from i to iii are declared outside the STUDENT independently. Show how to read and display member variables of DOB type if pointer variable is created for DOB inside STUDENT and STUDENT variable is also a pointer variable. The program should read and display the values of all members of STUDENT structure.

Filename: “student.h”

```
struct DOB
{
    int day;
    char* mth;
    int year;
};

struct STU_INFO
{
    int reg_no;
    char* name;
    char adrs[20];
};

struct COLLEGE
{
    char* clg_name;
    char univ_name[20] ;
};

struct STUDENT
{
    struct DOB *dob;
    struct STU_INFO stu_info;
    struct COLLEGE clg;
};
```

Filename: “lab2_q1.c”

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

```

#include "student.h"

int main()
{
    struct STUDENT *s1;
    s1=(struct STUDENT*)malloc(sizeof(struct STUDENT));
    s1->dob= (struct DOB*)malloc(sizeof(struct DOB));
    char month[10];
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");

    printf("Enter Student DOB day: ");
    scanf("%d", &s1->dob->day);
    printf("Enter Student DOB month: ");
    scanf("%s", month);
    printf("Enter Student DOB year: ");
    scanf("%d", &s1->dob->year);

    s1->dob->mth = (char*)calloc(strlen(month)+1,sizeof(char));
    strcpy(s1->dob->mth,month);

    char name[20];
    printf("Enter Student Registration Number: ");
    scanf("%d", &s1->stu_info.reg_no);
    printf("Enter Student Name: ");
    scanf("%s", name);
    printf("Enter Student Address: ");
    scanf("%s", s1->stu_info.adrs);
    s1->stu_info.name = (char*)calloc(strlen(name)+1,sizeof(char));
    strcpy(s1->stu_info.name,name);

    char college[30];
    printf("Enter Student College Name: ");
    scanf("%s", college);
    printf("Enter Student University Name: ");
    scanf("%s", s1->clg.univ_name);
    s1->clg.clg_name = (char*)calloc(strlen(college)+1,sizeof(char));
    strcpy(s1->clg.clg_name,college);

    printf("\nName: %s\n", s1->stu_info.name);
    printf("Registration Number: %d\n", s1->stu_info.reg_no);
    printf("Address: %s\n", s1->stu_info.adrs);
    printf("DOB : %d %s %d\n", s1->dob->day, s1->dob->mth, s1->dob->year);
    printf("College: %s\n", s1->clg.clg_name);
    printf("University: %s\n", s1->clg.univ_name);

    return 0;
}

```

```
Student@project-lab: ~/Udeet_200905406_C3/Week2
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week2$ gcc lab2_q1.c
Student@project-lab:~/Udeet_200905406_C3/Week2$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter Student DOB day: 27
Enter Student DOB month: 9
Enter Student DOB year: 2002
Enter Student Registration Number: 200905406
Enter Student Name: Udeet
Enter Student Address: Pune
Enter Student College Name: MIT
Enter Student University Name: MAHE

Name: Udeet
Registration Number: 200905406
Address: Pune
DOB : 27 9 2002
College: MIT
University: MAHE
Student@project-lab:~/Udeet_200905406_C3/Week2$ |
```

2. Write C program using recursion to copy one string to another using Recursion.

Filename: "lab2_q2.c"

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

void copy(char s1[], char s2[], int index)
{

s2[index] = s1[index];
if (s1[index] == '\0')
return;
copy(s1, s2, index + 1);

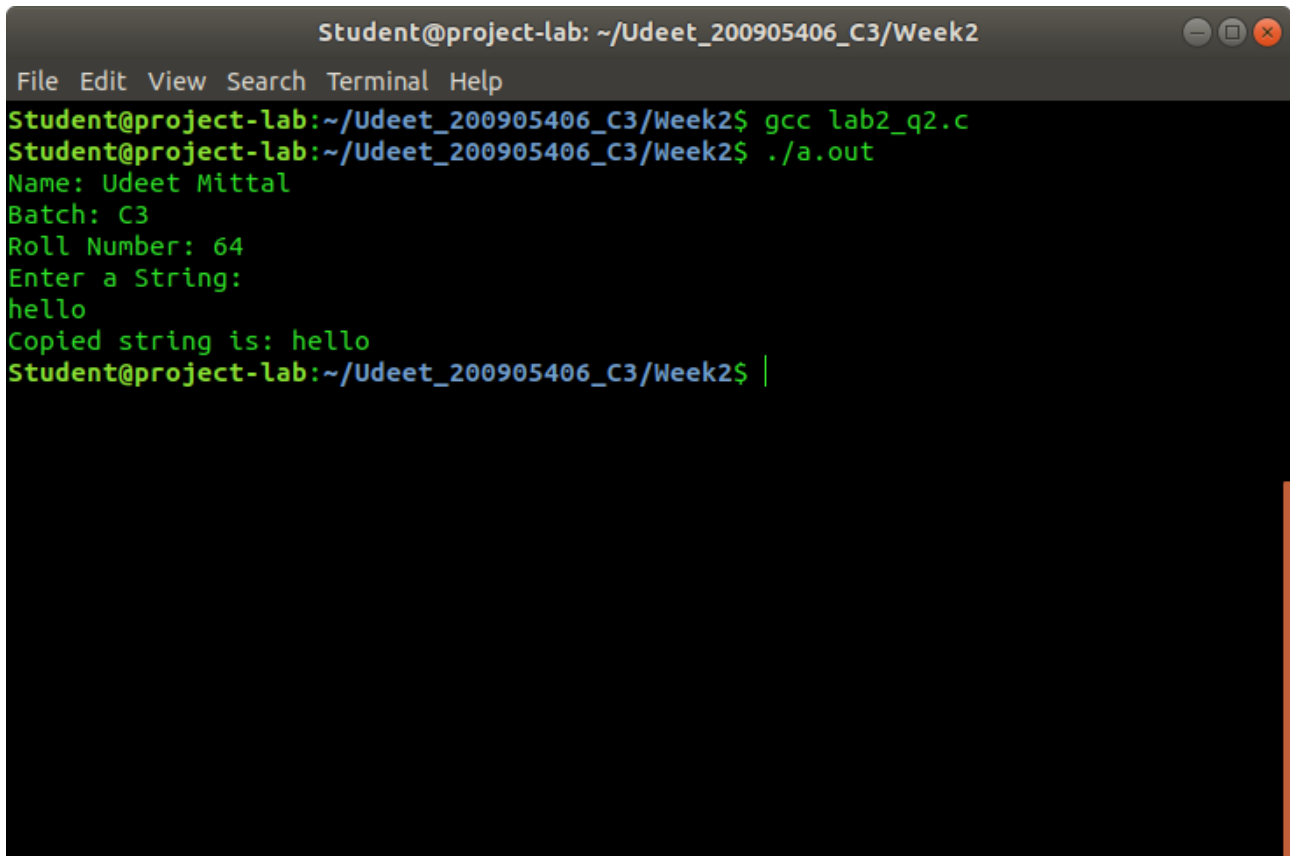
}

int main()
{

printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
```

```
char str[100];
printf("Enter a String:\n");
scanf("%s", str);
char copystr[100];
copy(str, copystr, 0);
printf("Copied string is: %s\n", copystr);

}
```

A screenshot of a terminal window titled "Student@project-lab: ~/Udeet_200905406_C3/Week2". The terminal shows the execution of a C program. The user enters "hello" in response to the prompt "Enter a String:". The program outputs "Copied string is: hello". The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command prompt is "Student@project-lab:~/Udeet_200905406_C3/Week2\$". The user enters "gcc lab2_q2.c" and then "./a.out". The program outputs "Name: Udeet Mittal", "Batch: C3", "Roll Number: 64", and "Enter a String:". The user enters "hello". The program outputs "Copied string is: hello". The terminal window has a dark background and green text. The user's input is shown in green, and the program's output is shown in green. The terminal window has a title bar with standard window controls (minimize, maximize, close). The terminal window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal window has a command prompt "Student@project-lab:~/Udeet_200905406_C3/Week2\$". The user enters "gcc lab2_q2.c" and then "./a.out". The program outputs "Name: Udeet Mittal", "Batch: C3", "Roll Number: 64", and "Enter a String:". The user enters "hello". The program outputs "Copied string is: hello". The terminal window has a dark background and green text. The user's input is shown in green, and the program's output is shown in green.

3. Write C programs using recursion to check whether a given String is Palindrome or not, using Recursion

Filename: "lab2_q3.c"

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

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

int checkpal(char str[], int s, int e)
{
    if (s >= e)
        return 1;

    if (str[s] != str[e])
        return 0;

    if (s < e + 1)
        return checkpal(str, s + 1, e - 1);

    return 1;
}

int main()
{
    printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
    char str[20];
    printf("Enter a string: ");
    scanf("%s", str);

    if(checkpal(str, 0, strlen(str)-1) == 1)
    {
        printf("The Entered String is Palindrome\n");
    }

    else
    {
        printf("The Entered String is not Palindrome\n");
    }

}
```

```
Student@project-lab: ~/Udeet_200905406_C3/Week2
File Edit View Search Terminal Help
Student@project-lab:~/Udeet_200905406_C3/Week2$ gcc lab2_q3.c
Student@project-lab:~/Udeet_200905406_C3/Week2$ ./a.out
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter a string: racecar
The Entered String is Palindrome
Student@project-lab:~/Udeet_200905406_C3/Week2$ |
```

4. Write C programs using recursion to simulate the working of Tower of Hanoi for n disks. Print the number of moves.

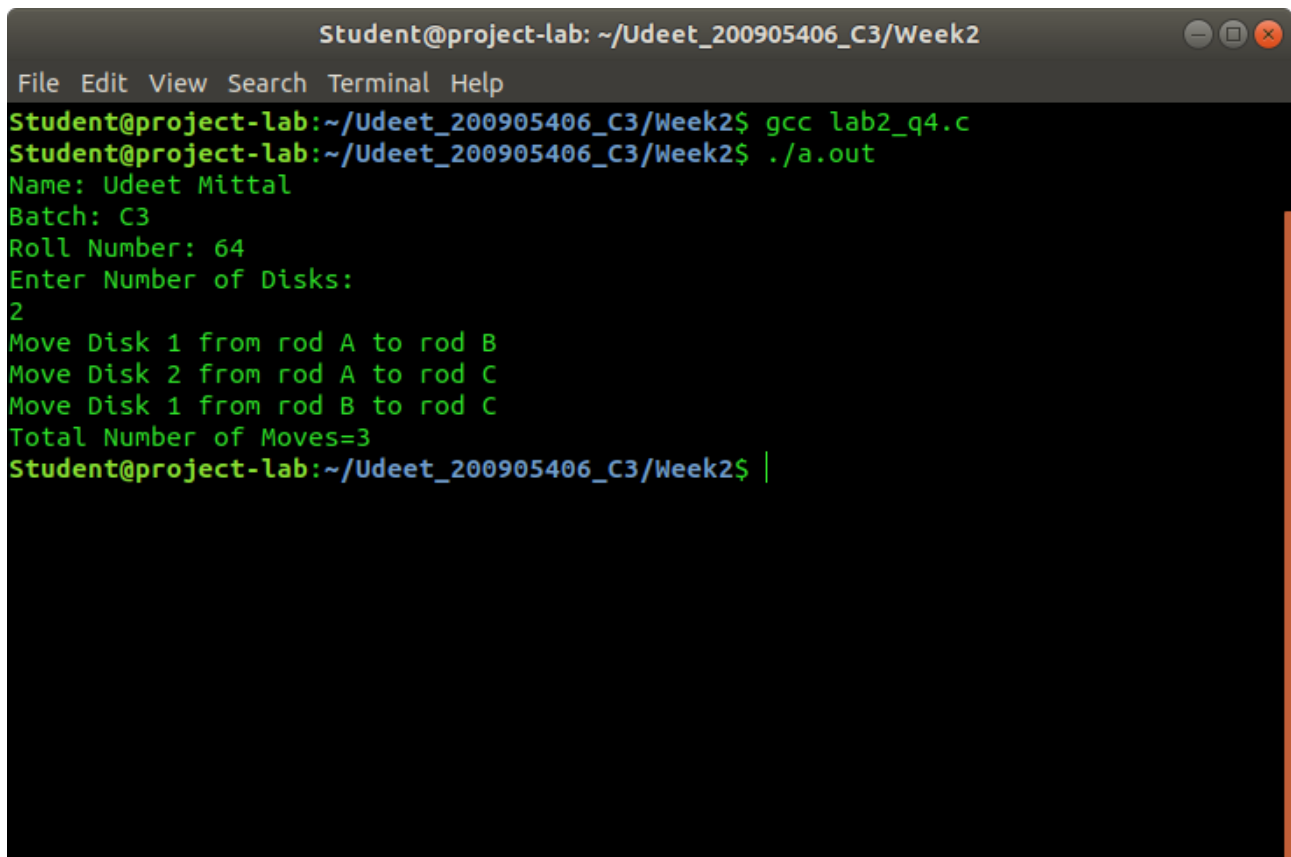
Filename: "lab2_q4.c"

```
#include <stdio.h>
#include <stdlib.h>
int TowerOfHanoi(int n, char from, char to, char inter)
{static int cnt=0;
if (n == 1)
{
printf("Move Disk 1 from rod %c to rod %c\n", from, to);
cnt++;
return cnt;
}

TowerOfHanoi(n - 1, from, inter, to);
printf("Move Disk %d from rod %c to rod %c\n", n, from, to);
cnt++;
TowerOfHanoi(n - 1, inter, to, from);
}
```



```
int main()
{
printf("Name: Udeet Mittal\nBatch: C3\nRoll Number: 64\n");
int n;
printf("Enter Number of Disks:\n");
scanf("%d", &n);
int cnt=TowerOfHanoi(n, 'A', 'C', 'B');
printf("Total Number of Moves=%d\n",cnt);
return 0;
}
```



The screenshot shows a terminal window titled "Student@project-lab: ~/Udeet_200905406_C3/Week2". The terminal has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The command prompt shows the user running the following commands:

```
Student@project-lab:~/Udeet_200905406_C3/Week2$ gcc lab2_q4.c
Student@project-lab:~/Udeet_200905406_C3/Week2$ ./a.out
```

The program's output is displayed in green text:

```
Name: Udeet Mittal
Batch: C3
Roll Number: 64
Enter Number of Disks:
2
Move Disk 1 from rod A to rod B
Move Disk 2 from rod A to rod C
Move Disk 1 from rod B to rod C
Total Number of Moves=3
Student@project-lab:~/Udeet_200905406_C3/Week2$ |
```