190905514 **Mohammad Tofik**

WEEK 2 LAB 2 :

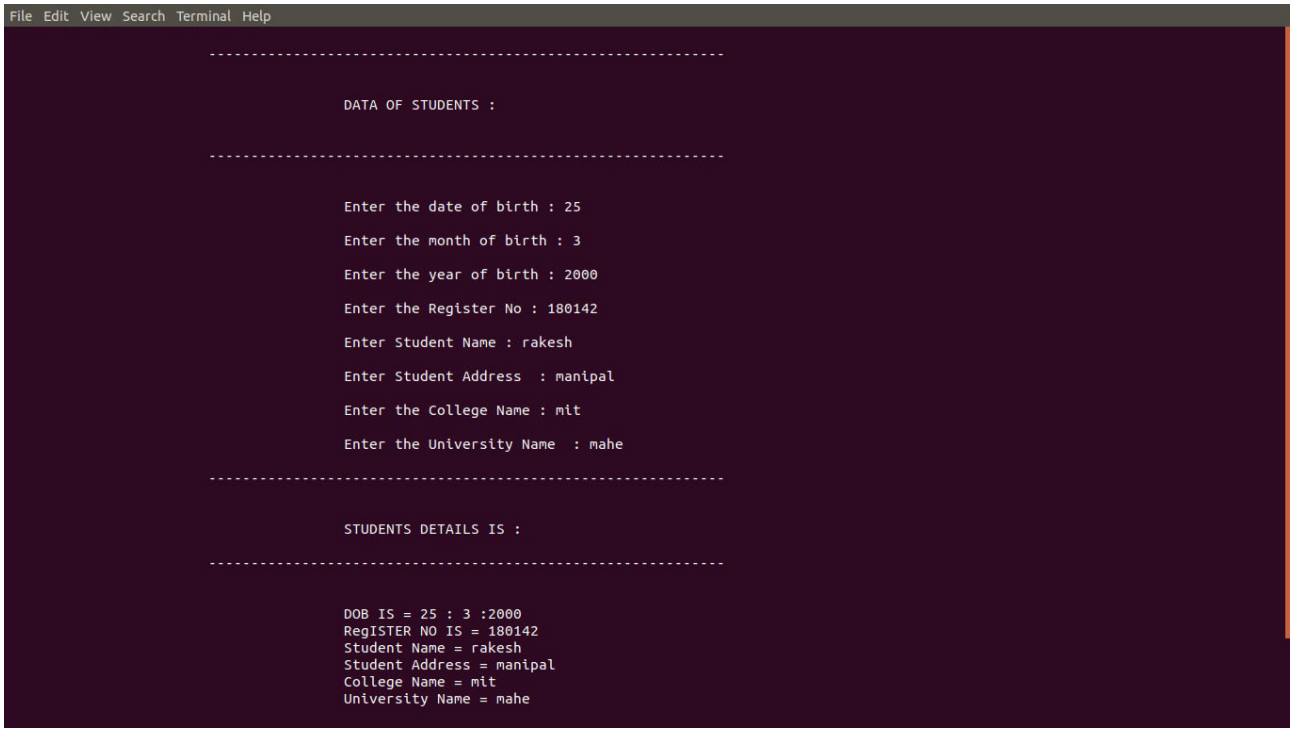
1. Create a structure STUDENT consisting of variables of structures:
i. DOB {day, month (use pointer), year},
ii. STU_INFO {reg_no, name(use pointer), address},
iii. COLLEGE {college_name (use pointer), university_name}
where structure types from i to iii are declared outside the STUDENT independently.
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.

program1.c

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct DOB {
int day;
char *month;
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;
};


int main(){
struct STUDENT *s;
char n[20],c[20],m[20];
s = (struct STUDENT*)malloc(sizeof(struct STUDENT));
s->dob = (struct DOB*)malloc(sizeof(struct DOB));
printf("\n\t\t\t----------------------------------------------------------\n\n");
printf("\n\t\t\t\t\tDATA OF STUDENTS : \n\n");
printf("\n\t\t\t----------------------------------------------------------\n\n");
printf("\n\t\t\t\t\tEnter the date of birth : ");
scanf("%d",&s->dob->day);
printf("\n\t\t\t\t\tEnter the month of birth : ");
scanf("%s",m);
printf("\n\t\t\t\t\tEnter the year of birth : ");
```

```
scanf("%d",&s->dob->year);
s->dob->month = (char*)calloc(strlen(m)+1,sizeof(char));
strcpy(s->dob->month,m);
printf("\n\t\t\t\t\tEnter the Register No : ");
scanf("%d",&s->stu_info.reg_no);
printf("\n\t\t\t\t\tEnter Student Name : ");
scanf("%s",n);
s->stu_info.name = (char*)calloc(strlen(n)+1,sizeof(char));
strcpy(s->stu_info.name,n);
printf("\n\t\t\t\t\tEnter Student Address  : ");
scanf("%s",s->stu_info.adrs);
printf("\n\t\t\t\t\tEnter the College Name : ");
scanf("%s",c);
s->clg.clg_name = (char*)calloc(strlen(c)+1,sizeof(char));
strcpy(s->clg.clg_name,c);
printf("\n\t\t\t\t\tEnter the University Name  : ");
scanf("%s",s->clg.univ_name);
printf("\n\t\t\t----------------------------------------------------------------\n\n");
printf("\n\t\t\t\t\tSTUDENTS DETAILS IS : \n");
printf("\n\t\t\t----------------------------------------------------------------\n\n");
printf("\n\t\t\t\t\tDOB IS = %d : %s :%d ",s->dob->day,s->dob->month,s->dob->year);
printf("\n\t\t\t\t\tRegISTER NO IS = %d ",s->stu_info.reg_no);
printf("\n\t\t\t\t\tStudent Name = %s ",s->stu_info.name);
printf("\n\t\t\t\t\tStudent Address = %s",s->stu_info.adrs);
printf("\n\t\t\t\t\tCollege Name = %s",s->clg.clg_name);
printf("\n\t\t\t\t\tUniversity Name = %s\n",s->clg.univ_name);
printf("\n\t\t\t----------------------------------------------------------------\n\n");
}
```

OUTPUT :



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

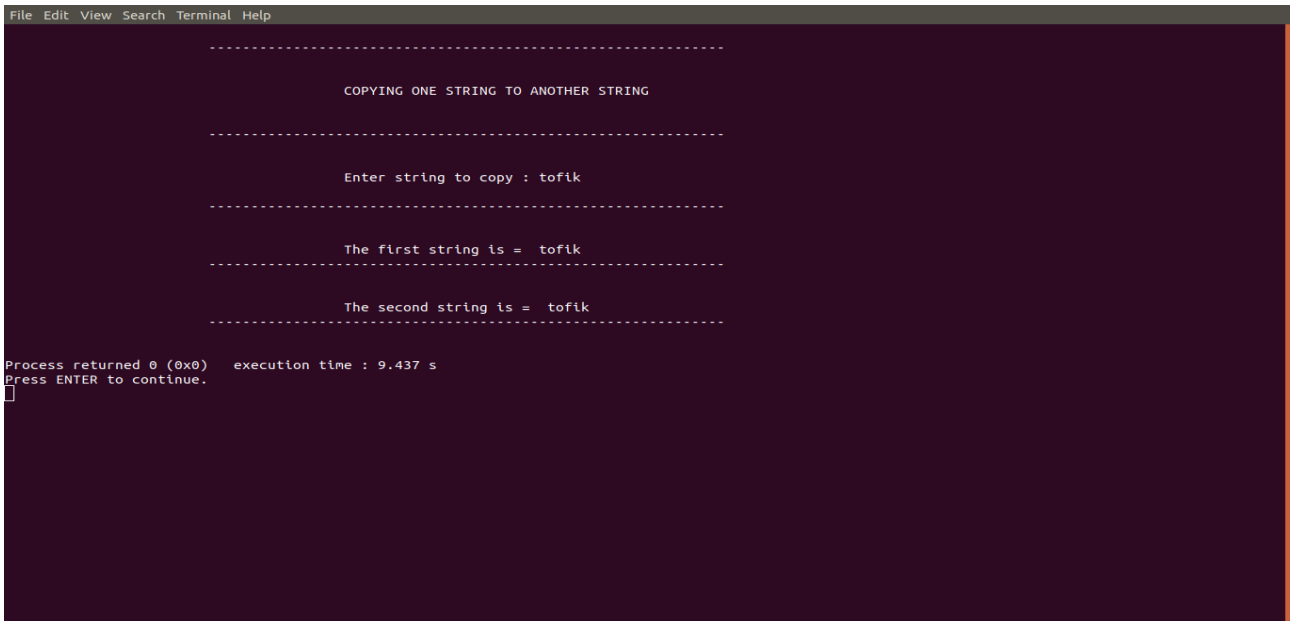
function_string_copy.h

```
void copystring(char string1[], char string2[], int index_d)
{
   string2[index_d] = string1[index_d];
   if (string1[index_d] == '\0')
      return;
   copystring(string1, string2, index_d + 1);
}
```

program2.c

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

void copy(char [], char [], int);

int main(void)
{
   char string1[40], string2[40];
   printf("\n\t\t\t----------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tCOPYING ONE STRING TO ANOTHER STRING \n\n");
   printf("\n\t\t\t----------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tEnter string to copy : ");
   scanf("%s", string1);
   printf("\n\t\t\t----------------------------------------------------------\n\n");
   copystring(string1, string2, 0);
   printf("\n\t\t\t\t\tThe first string is =  %s ", string1);
   printf("\n\t\t\t----------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tThe second string is =  %s ", string2);
   printf("\n\t\t\t----------------------------------------------------------\n\n");

}
```

OUTPUT :



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

function_palindrome.h

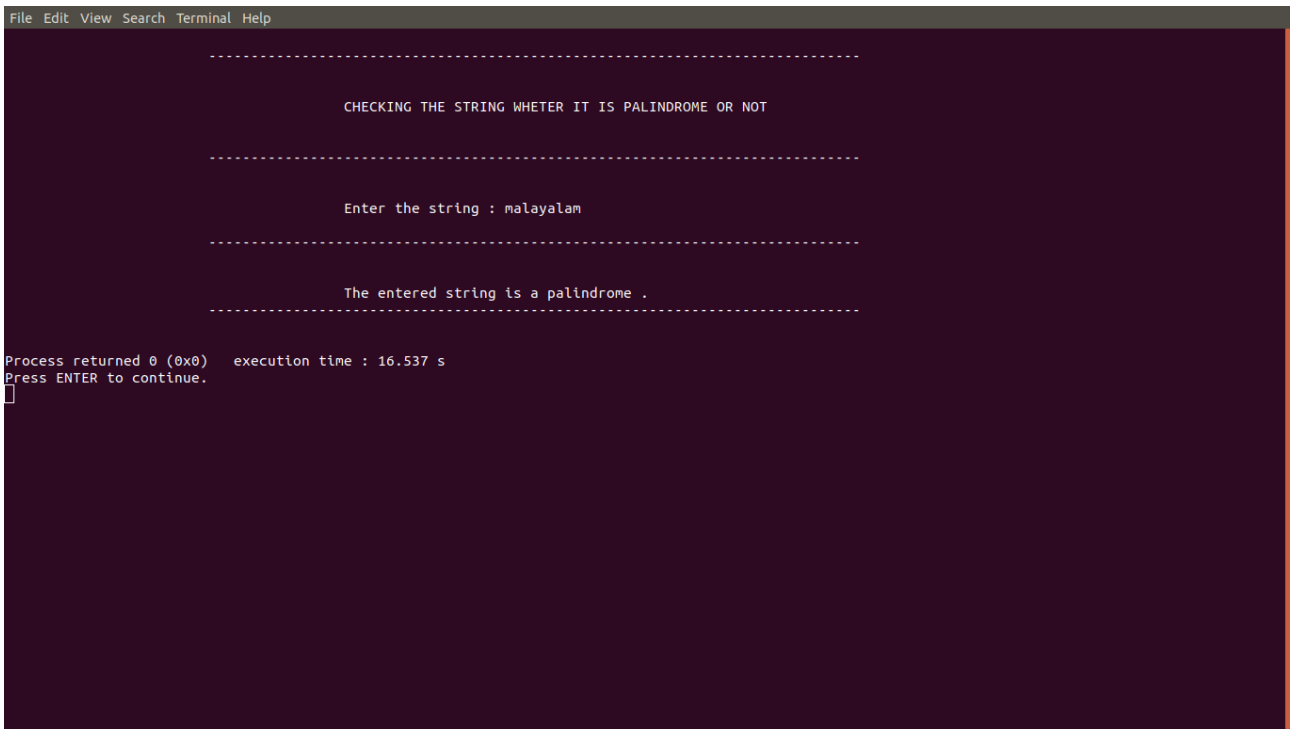
```
void palindrome(char ch[], int index_d)
{
   int length = strlen(ch) - (index_d + 1);
   if (ch[index_d] == ch[length])
   {
      if (index_d + 1 == length || index_d == length)
      {   printf("\n\t\t\t----------\n\n");
          printf("\n\t\t\t\t\tThe entered string is a palindrome . ");
          printf("\n\t\t\t----------\n\n");
          return;
      }
      palindrome(ch, index_d + 1);
   }
   else
   {   printf("\n\t\t\t----------\n\n");
      printf("\n\t\t\t\t\tThe entered string is not a palindrome : ");
      printf("\n\t\t\t----------\n\n");
   }
```

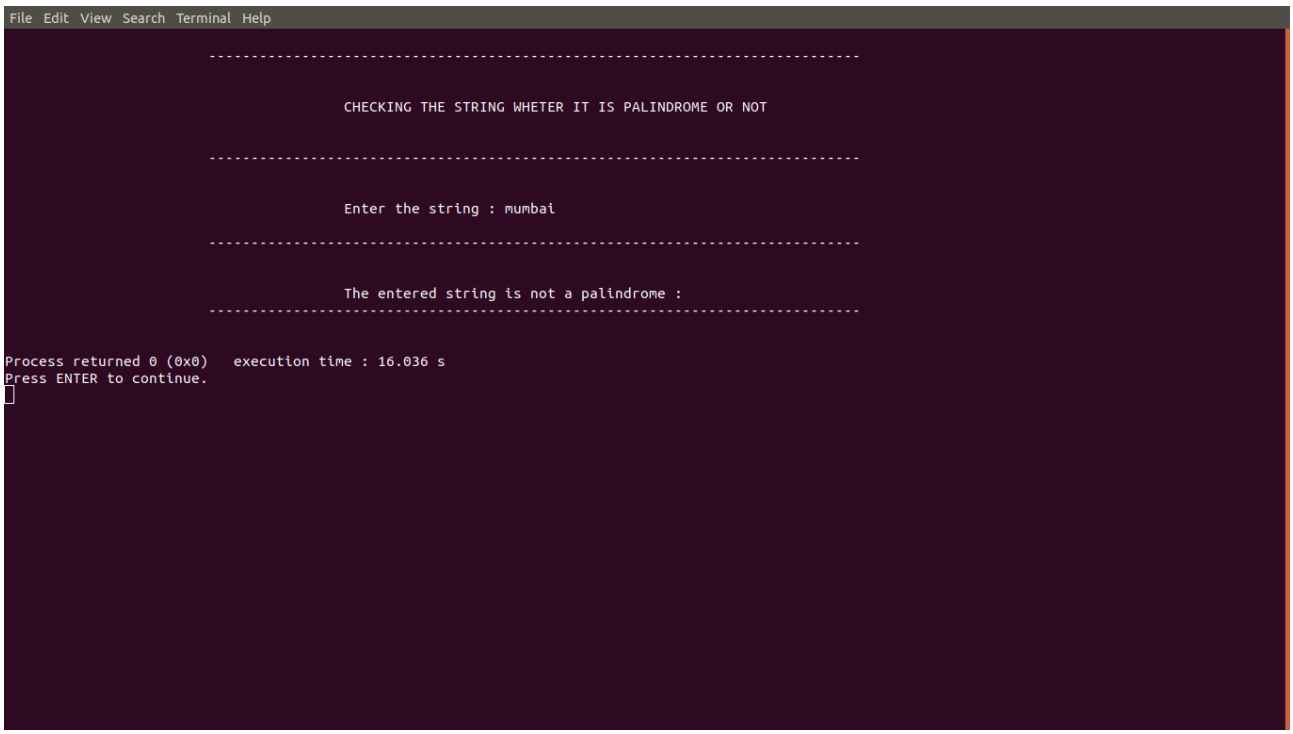
```
}
```

program3.c

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


int main(void)
{
    char ch[25];

    printf("\n\t\t\t----------------------------------------------------------------------------\n\n");
    printf("\n\t\t\t\t\tCHECKING THE STRING WHETER IT IS PALINDROME OR NOT\n\n");
    printf("\n\t\t\t----------------------------------------------------------------------------\n\n");
    printf("\n\t\t\t\t\tEnter the string : ");
    scanf("%s", ch);
    palindrome(ch, 0);

}
```

OUTPUT :


File Edit View Search Terminal Help
CHECKING THE STRING WHETER IT IS PALINDROME OR NOT
Enter the string : malayalam
The entered string is a palindrome .
Process returned 0 (0x0) execution time : 16.537 s
Press ENTER to continue.


File Edit View Search Terminal Help
CHECKING THE STRING WHETER IT IS PALINDROME OR NOT
Enter the string : mumbai
The entered string is not a palindrome :
Process returned 0 (0x0) execution time : 16.036 s
Press ENTER to continue.

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

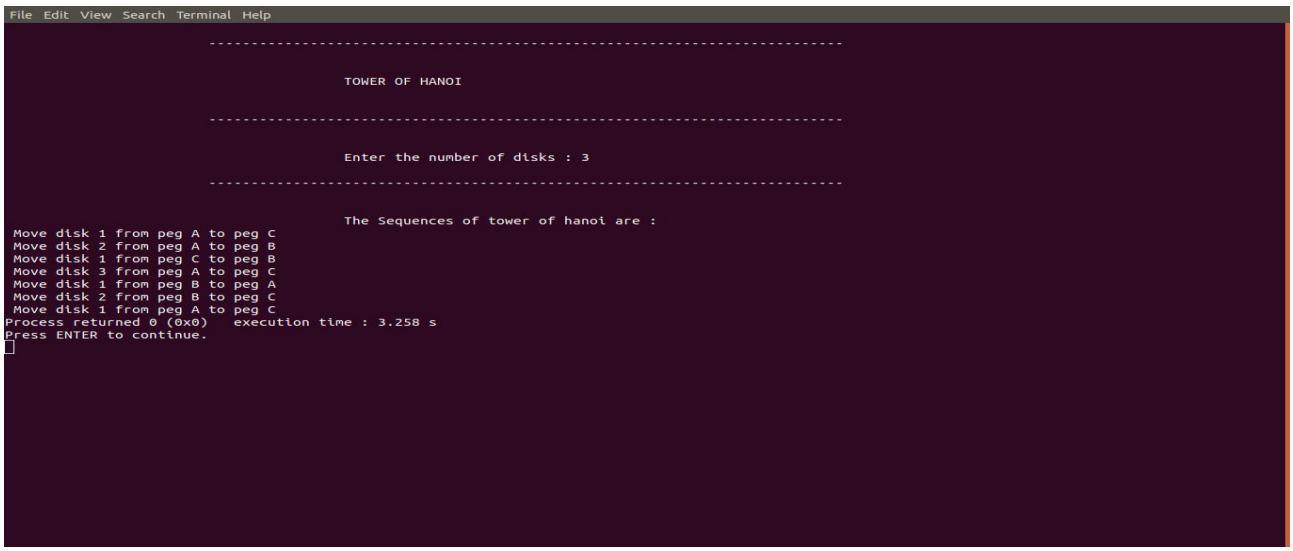
function_towerofhanoi.h

```
void towerofhanoi(int number, char disk, char topdisk, char axulliary)
{
   if (number == 1)
   {
      printf("\n Move disk 1 from peg %c to peg %c", disk, topdisk);
      return;
   }
   towerofhanoi(number - 1, disk, axulliary, topdisk);
   printf("\n Move disk %d from peg %c to peg %c", number, disk, topdisk);
   towerofhanoi(number - 1, axulliary, topdisk, disk);

}
```

program4.c

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

int main(void)
{
   int number;
   printf("\n\t\t\t----------------------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tTOWER OF HANOI \n\n");
   printf("\n\t\t\t----------------------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tEnter the number of disks : ");
   scanf("%d", &number);
   printf("\n\t\t\t----------------------------------------------------------------------\n\n");
   printf("\n\t\t\t\t\tThe Sequences of tower of hanoi are : ");
   towerofhanoi(number, 'A', 'C', 'B');


}
```

OUTPUT :


File Edit View Search Terminal Help
TOWER OF HANOI
Enter the number of disks : 3
The Sequences of tower of hanoi are :
Move disk 1 from peg A to peg C
Move disk 2 from peg A to peg B
Move disk 1 from peg C to peg B
Move disk 3 from peg A to peg C
Move disk 1 from peg B to peg A
Move disk 2 from peg B to peg C
Move disk 1 from peg A to peg C
Process returned 0 (0x0) execution time : 3.258 s
Press ENTER to continue.