

**WEEK-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.**

**Program:**

```
#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("Enter Day Month and Year of Birth seperately:");
    scanf("%d",&s->dob->day);
    scanf("%s",m);
    scanf("%d",&s->dob->year);
    s->dob->month = (char*)calloc(strlen(m)+1,sizeof(char));
    strcpy(s->dob->month,m);
    printf("Reg No.: ");
```

```

scanf("%d",&s->stu_info.reg_no);
printf("Student Name: ");
scanf("%s",n);
s->stu_info.name = (char*)calloc(strlen(n)+1,sizeof(char));
strcpy(s->stu_info.name,n);
printf("Address: ");
scanf("%s",s->stu_info.adrs);
printf("College Name: ");
scanf("%s",c);
s->clg.clg_name = (char*)calloc(strlen(c)+1,sizeof(char));
strcpy(s->clg.clg_name,c);
printf("University Name: ");
scanf("%s",s->clg.univ_name);
printf("\nStudent Info: \n");
printf("DOB: %d:%s:%d \n",s->dob->day,s->dob->month,s->dob->year);
printf("Reg No.: %d\n",s->stu_info.reg_no);
printf("Student Name: %s\n",s->stu_info.name);
printf("Student Address: %s\n",s->stu_info.adrs);
printf("College Name: %s\n",s->clg.clg_name);
printf("University Name: %s\n",s->clg.univ_name);
}

```

#### Test Case:

```

Enter Day Month and Year of Birth seperately: 20 02 2000
Reg No.: 190905513
Student Name: Danish
Address: XBlock
College Name: MIT
University Name: MAHE

Student Info:
DOB: 20:02:2000
Reg No.: 190905513
Student Name: Danish
Student Address: XBlock
College Name: MIT
University Name: MAHE

Process returned 0 (0x0)    execution time : 23.845 s
Press ENTER to continue.

```

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

### Program:

```
#include<stdio.h>
#include<string.h>
void copy(char a[],char b[],int i)
{
    if(a[i]=='\0')
    {
        b[i] = '\0';
        return;
    }
    b[i] = a[i];
    copy(a,b,i+1);
}

int main()
{
    char a[20],b[20];
    printf("Enter the string: ");
    scanf("%s",a);
    copy(a,b,0);
    printf("String after copying: %s\n",b);
    return 0;
}
```

### Test Case:

```
Enter the string: Manipal190905513
String after copying: Manipal190905513

Process returned 0 (0x0)    execution time : 12.243 s
Press ENTER to continue.
█
```

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

#### Program:

```
#include<stdio.h>
#include<string.h>
void palindrome(char a[],int n,int i){
    if(i>=n-i-1){
        printf("The word %s is a Palindrome\n",a);
        return;
    }
    if(a[i]!=a[n-i-1]){
        printf("The word %s is not a Palindrome\n",a);
        return;
    }
    palindrome(a,n,i+1);
}

int main(){
    char a[20];
    int n;
    printf("Enter the string to check whether its palindrome or not: ");
    scanf("%s",a);
    n = strlen(a);
    palindrome(a,n,0);
    return 0;
}
```

#### Test Case:

```
Enter the string to check whether its palindrome or not: Danish
The word Danish is not a Palindrome

Process returned 0 (0x0)   execution time : 11.939 s
Press ENTER to continue.
█
```

```
Enter the string to check whether its palindrome or not: mom
The word mom is a Palindrome

Process returned 0 (0x0)   execution time : 12.018 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.**

**Program:**

```
#include<stdio.h>
int tower(int n , char source , char temp , char dest){
    static int count = 0;
    if(n==1){
        printf("Moved Disk 1 From %c To %c \n",source,dest);
        count++;
        return count;
    }
    tower(n-1,source,dest,temp);
    printf("Moved Disk %d From %c To %c \n",n,source,dest);
    count++;
    tower(n-1,temp,source,dest);
}
void main(){
    char a,b,c;
    int n;
    printf("Enter the number of disks:");
    scanf("%d",&n);

    printf("The total moves are: %d\n", tower(n,'a','b','c'));
}
```

**Test Case:**

```
Enter the number of disks:3
Moved Disk 1 From a To c
Moved Disk 2 From a To b
Moved Disk 1 From c To b
Moved Disk 3 From a To c
Moved Disk 1 From b To a
Moved Disk 2 From b To c
Moved Disk 1 From a To c
The total moves are: 7

Process returned 0 (0x0)    execution time : 2.174 s
Press ENTER to continue.
█
```

Enter the number of disks:4

Moved Disk 1 From a To b

Moved Disk 2 From a To c

Moved Disk 1 From b To c

Moved Disk 3 From a To b

Moved Disk 1 From c To a

Moved Disk 2 From c To b

Moved Disk 1 From a To b

Moved Disk 4 From a To c

Moved Disk 1 From b To c

Moved Disk 2 From b To a

Moved Disk 1 From c To a

Moved Disk 3 From b To c

Moved Disk 1 From a To b

Moved Disk 2 From a To c

Moved Disk 1 From b To c

The total moves are: 15

Process returned 0 (0x0)

execution time : 3.094 s

Press ENTER to continue.

■