WEEK 1:

a) Introduction to structures in "C"

AIM: To write a C program to find the total, average of n students using structures.

DESCRIPTION:

- A structure is a user defined data type that stores data of different types under one name.
- Structure variables are declared using "struct" keyword together with structure name and the data held by the structure.
- Individual data items stored under a structure is known as members. These can be of basic data types like int, char, double or derived data types like arrays, strings etc.

Syntax:

```
struct [structure name]
{
        datatype member1;
        datatype member2;
        ……..
        datatype member n;
}structure1,structure2, … ,structure n;
```

- You can declare a structure variable during creation of structure template as shown above.
    - Structure1, structure2, structure n are variables of type [structure name] structure.
- You can also declare the structure variable separately as follows:
    - `struct student student1; //creates a structure student1`
- Structure variables are accessed using the structure name together with (.)dot operator as shown below.
    - `stud.fees = 123.4; // sets values of fees member to 123.4.`

ALGORITHM

STEP 1: Start

STEP 2: Create student structure, stud

STEP 3: Read number of students, n

STEP 4: (i=0 ; i<n ; i++), goto 5, else goto 6

```
STEP 5: a) Read student[i+1]'s name
    b) Read student[i+1]'s roll number
    c) (j=0 ; j<4 ; j++)
            Read student[i]'s 4 marks
STEP 6: (i=0 ; i<n ; i++) , goto a, else goto 7
       a) total = 0;
       b) (j=0 ; j<4 ; j++)
            total = total + stud[i].marks[j];
            Output total
            Calculate average = total/4.0;
STEP 7: (i=0 ; i<n ; i++), goto a, else goto 8
       a) Output average for i'th student
STEP 8: Stop
```

PROGRAM:

```
#include <stdio.h>
int main()
{
        typedef struct student{
                int roll;
                char name[100];
                int marks[4];
                float avg;
        }student;
        student stud[100];
        int n;
        printf("Enter Number Of Students:");
        scanf("%d",&n);
        for(int i=0;i<n;i++){   printf("\nEnter Name for student %d\n:",i+1);
                scanf("%s",&stud[i].name);
                fflush(stdin);
                printf("\nEnter Roll Number for student %d: ",i+1);
```

```
            scanf("%d",&stud[i].roll);
            fflush(stdin);
            printf("\nEnter marks for 4 subjects for student %d: ",i+1);
                for(int j=0;j<4;j++){
                scanf("%d",&stud[i].marks[j]);
            }
        }
        //Calculattions
        for(int i=0;i<n;i++){
            int total=0;
            for(int j=0;j<4;j++){
                total += stud[i].marks[j];
            }
            printf("\nThe total for %s = %d: ",stud[i].name,total);
            stud[i].avg = (float)total/4;
        }
        //Output
        for(int i =0;i<n;i++){
            printf("\nAverage for %s = %.2f",stud[i].name,stud[i].avg);
        }
        return 0;
}
```

OUTPUT:

```
Enter Number Of Students:3
Enter Name for student 1 :Trinity
Enter Roll Number for student 1: 2413
Enter marks for 4 subjects for student 1: 20 40 60 80

Enter Name for student 2 :Darlington
```

- To confirm this, we check the contents of stud1[100], by using the (.) operator with the help of structure name and index.

ALGORITHM:

```
STEP 1: Start

STEP 2: Create student structure, stud

STEP 3: Read number of students, n

STEP 4: (i=0 ; i<n ; i++), goto 5, else goto 6

STEP 5: a) Read student[i+1]'s name

   d) Read student[i+1]'s roll number
   e) (j=0 ; j<4 ; j++)
            Read student[i]'s 4 marks

STEP 6: (i=0 ; i<n ; i++) , goto a, else goto 7

      c) total = 0;
      d) (j=0 ; j<4 ; j++)
            total = total + stud[i].marks[j];
            Output total
            Calculate average = total/4.0;

STEP 7: (i=0 ; i<n ; i++), goto a, else goto 8

      c) Output average for i'th student

STEP 8: (i=0 ; i<n ; i++), copy stud[i] to stud1[i]

STEP 9: Output contents of stud1

STEP : Stop
```

PROGRAM:

```
#include <stdio.h>
int main(){
    typedef struct student{
        int roll;
        char name[100];
        int marks[4];
        float avg;
    }student;
```

```
    student stud[100];
    student stud1[100];
    int n;
    printf("Enter Number Of Students:");
    scanf("%d",&n);
    for(int i=0;i<n;i++){
            printf("\nEnter Name for student %d\n:",i+1);
            scanf("%s",&stud[i].name);
            fflush(stdin);

            printf("\nEnter Roll Number for student %d: ",i+1);
            scanf("%d",&stud[i].roll);
            fflush(stdin);
            printf("\nEnter marks for 4 subjects for student %d: ",i+1);
                    for(int j=0;j<4;j++){
                            scanf("%d",&stud[i].marks[j]);
                    }
  }
  //Calculattions
  for(int i=0;i<n;i++){
          int total=0;
          for(int j=0;j<4;j++){
                  total += stud[i].marks[j];
          }
          printf("\nThe total for %s = %d: ",stud[i].name,total);
          stud[i].avg = (float)total/4;
}
//Output
for(int i =0;i<n;i++){
        printf("\nAverage for %s = %.2f",stud[i].name,stud[i].avg);
```

Enter Roll Number for student 2: 4565

Enter marks for 4 subjects for student 2: 90 90 90 100

Enter Name for student 3 :Makarios

Enter Roll Number for student 3: 9898

Enter marks for 4 subjects for student 3: 100 100 100 50

The total for Trinity = 200:

The total for Darlington = 370:

The total for Makarios = 350:

Average for Trinity = 50.00

Average for Darlington = 92.50

Average for Makarios = 87.50

WEEK 1

b) Copy one structure variable to another structure of the same type.

AIM: Writing a C program to copy one structure variable to another structure of the same type.

DESCRIPTION:

- You can copy contents of one structure variable to another structure variable of the same type by using the assignment operator(=).
- All members of the structure are copied to the destination structure together with their values.
- **SYNTAX:** `stud2 = stud1;`
- In this program, we create a structure variable: `struct student stud[100];`
- We create another structure variable struct student stud2; which is the destination structure.
- After populating the members of stud[100], we then copy it to stud1[100];

```
    }
    //Copying
    for(int i=0;i<n;i++)
    stud1[i] = stud[i];
    printf("\nsizeof structure 2 =%d bytes ",sizeof(stud1));
    return 0;
}
```

OUTPUT:

```
Enter Number Of Students:2
Enter Name for student 1 :Trinity
Enter Roll Number for student 1: 1111
Enter marks for 4 subjects for student 1: 50 60 70 80

Enter Name for student 2 :Andra
Enter Roll Number for student 2: 10101
Enter marks for 4 subjects for student 2: 30 30 30 50

The total for Trinity = 260:
The total for Andra = 140:

Average for Trinity = 65.00
Average for Andra = 35.00
sizeof structure 2 = 12400 bytes
```