Date-

Assignment No. :

Problem Statement:

Program in C to calculate the average and highest marks among given students.

Theory:

A structure is a user defined data type in C. A structure creates a data type that can be used to group items of possibly different types into a single type. 'struct' keyword is used to create a structure. A structure variable can either be declared with structure declaration or as a separate declaration like basic types. It can be initialized using curly braces '{}' not with declaration. Structure members are accessed using dot (.) operator. Like other primitive data types, we can create an array of structures as we used in this program.

Here in this program, a structure 'stud' is used to have a record of a student's name, roll and marks in two subjects(e.g.: 'subject 1' and 'subject 2'). Now, after having the records the program will be able to calculate the average marks in each subject and to show the roll and name of the students who have got the highest marks in each subject.

Algorithm:

Input specification: 1: An array s[i], of a user defined structure consists of 3 integer variables: roll, marks1, marks2 and a character type array: name[].

2: Number of students n.

Output specification: Details, Highest marks and average marks of the students.

Steps:

Algorithm for method main():

Step 1: Print "Enter number of students: "
Step 2: Input n
Step 3: Repeat step 4 For i = 0 to n-1
Step 4: entry(i)
Step 5: i=i+1
[End of For]
Step 6: Repeat through step7 to step 17 while(ch!=4)
Step 7: Print"1. Details of student
2. Average marks
3. Highest marks
4. Exit"
Step 8: Print ”Enter your choice : "
Step 9: Input ch
Step 10: If(ch= 1) Then
Step 11: details()
Step 12: Else If(ch= 2) Then
Step 13: avg()
Step 14: Else If(ch=3) Then
Step 15: highest()
Step 16: Else If(ch!=4)
Step 17: Print "Wrong choice!"
[End of If structure]
[End of While loop]
Step 18: Stop
[End of method main()]

Algorithm for method entry(i):

Step 1: j=0
Step 2: Print "Enter the details of student "i+1”: “
Step 3: Print ”Enter roll: "
Step 4: Input s[i].roll

Step 5: Print "Enter name: "
Step 6: Input s[i].name
Step 7: Print "Enter marks in subject 1: "
Step 8: Input s[i].marks1
Step 9: Print "Enter marks in subject 2: "
Step 10: Input s[i].marks2
Step 11: Stop
[End of method entry()]

Algorithm for method details():

Step 1: Print "Enter the roll no: "
Step 2: Input r
Step 3: Repeat through step 4 to step 11 For i = 0 to n-1
Step 4: If(s[i].roll = r) Then
Step 5: Print "Details of student:"
Step 6: Print "Name: "s[i].name
Step 7: Print "Marks in:"
Step 8: Print "Subject 1: "s[i].m1
Step 9: Print "Subject 2: "s[i].m2
Step 10: Return
[End of If]
Step 11: i=i+1
[End of For loop]
Step 12: If(i = n) Then
Step 13: Print "Record does not exist for roll " r
[End of If structure]
Step 14: Stop
[End of method details()]

Algorithm for method avg():

Step 1: t1 = 0
Step 2: t2 = 0
Step 3: Repeat through step 4 to step 6 For i = 0 to n-1
Step 4: t1=t1+ s[i].m1
Step 5: t2=t2+s[i].m2
Step 6: i=i+1
[End of For loop]

Step 7: avg1 = t1/n
Step 8: avg2 = t2/n
Step 9: Print "Average marks in sub 1: "avg1
Step 10: Print "Average marks in sub 2: "avg2
Step 11: Stop
[End of method avg()]

Algorithm for method highest():

Step 19: max1=0
Step 20: max2=0
Step 21: Repeat through step 4 to step 10 For i = 0 to n-1
Step 22: If(s[i].m1 > max1)
Step 23: max1 = s[i].m1
Step 24: t1 = i
[End of If structure]
Step 25: If(s[i].m2 > max2)
Step 26: max2 = s[i].m1
Step 27: t2 = i
[End of If structure]
Step 28: i=i+1
[End of For loop]
Step 29: Print "Details of student(s) who scored the highest marks:"
Step 30: If(t1=t2)
Step 31: Print "Roll: "s[t1].roll "Name: "s[t1].name
Step 32: Print "Scored highest marks in both the subjects"
Step 33: Else
Step 34: Print "Roll: "s[t1].roll "Name: "s[t1].name
Step 35: Print "Scored the highest marks in subject 1"
Step 36: Print "Roll: "s[t2].roll "Name: " s[t2].name
Step 37: Print "Scored the highest marks in subject 2"
Step 38: [End of If-Else structure]
Step 39: Stop
[End of method highest()]

Source Code:

```
#include<stdio.h>
struct student{
        int roll;
        char name[30];
        int m1, m2;
}s[20];
void entry(int);
void avg();
void highest();
void details();
int n, i;
int main()
{
        int ch;
        printf("Enter number of students: ");
        scanf("%d", &n);
        for(i = 0;i < n;i++)
                entry(i); // Calling method entry
        while(1)
        {
                printf("\n1. Details of student\n2. Average marks\n3. Highest
                                marks\n4. Exit");
                printf("\nEnter your choice : ");
                scanf("%d", &ch);
                switch(ch)
                {
                        case 1:         details(); // Find the details of a particular student
                                        break;
                        case 2:         avg(); // Displaying average marks
                                        break;
                        case 3:         highest(); //Displaying highest marks in each
                                                        //subject
                                        break;
                        case 4:         return 0;
```

```
                        default:        printf("\nWrong choice!");
                }
        }
        return 0;
}
void entry(int i) // Definition of 'entry'
{
        int j=0;
        printf("\nEnter the details of student %d :", i+1);
        printf("\nEnter roll: ");
        scanf("%d", &s[i].roll);
        printf("\nEnter name: ");
        fflush(stdin);
        gets(s[i].name);
        printf("\nEnter marks in subject 1: ");
        scanf("%d", &s[i].m1);
        printf("\nEnter marks in subject 2: ");
        scanf("%d", &s[i].m2);
}
void details() // Definition of 'display'
{
        int r;
        printf("\nEnter the roll no: ");
        scanf("%d", &r);
        for(i = 0;i <  n;i++)
        {
                if(s[i].roll == r)
                {
                        printf("\nDetails of student:");
                        printf("\n\tName: %s",s[i].name);
                        printf("\nMarks in:");
                        printf("\n\tSubject 1: %d", s[i].m1);
                        printf("\n\tSubject 2: %d", s[i].m2);
                        break;
                }
        }
        if(i == n)
```

```
                printf("\nRecord does not exist for roll %d", r);
}

void avg()// Definition of method average
{
        int t1 = 0, t2 = 0;
        float avg1, avg2;
        for(i = 0;i<n;)
        {
                t1 += s[i].m1; // Finding sum of all marks in sub1
                t2 += s[i++].m2; // Finding sum of all marks in sub2
        }
        avg1 = t1/n; avg2 = t2/n;
        printf("\nAverage marks in sub 1: %0.2f", avg1);
        printf("\nAverage marks in sub 2: %0.2f", avg2);
}
void highest()//Definition of method highest
{
        int max1 = 0, max2 = 0, t1, t2;
        for(i = 0;i < n;i++)
        {
                if(s[i].m1 > max1)
                {
                        max1 = s[i].m1;
                        t1 = i;
                }
                if(s[i].m2 > max2)
                {
                        max2 = s[i].m1;
                        t2 = i;
                }
        }
        printf("\nDetails of student(s) who scored the highest marks:");
        if(t1 == t2)//when a particular student has achieved highest marks in both
                        //the subjects
        {
                printf("\n\tRoll: %d\n\tName: %s", s[t1].roll, s[t1].name);
```

```
            printf("\nScored highest marks in both the subjects");
        }
        Else
        {
            printf("\n\tRoll: %d\n\tName: %s", s[t1].roll, s[t1].name);
            printf("\nScored the highest marks in subject 1");
            printf("\n\tRoll: %d\n\tName: %s", s[t2].roll, s[t2].name);
            printf("\nScored the highest marks in subject 2");
        }
}
```

Input & Output:

```
Enter number of students: 2

Enter the details of student 1 :
Enter roll: 1

Enter name: John Kits

Enter marks in subject 1: 60

Enter marks in subject 2: 70

Enter the details of student 2 :
Enter roll: 2

Enter name: Terry Jane

Enter marks in subject 1: 80

Enter marks in subject 2: 50

1. Details of student
2. Average marks
3. Highest marks
4. Exit
Enter your choice : 1

Enter the roll no: 1

Details of student:
        Name: John Kits
Marks in:
        Subject 1: 60
        Subject 2: 70
1. Details of student
2. Average marks
3. Highest marks
4. Exit
```

```
Enter your choice : 2

Average marks in sub 1: 70.00
Average marks in sub 2: 60.00
1. Details of student
2. Average marks
3. Highest marks
4. Exit
Enter your choice : 3

Details of student(s) who scored the highest marks:
        Roll: 2
        Name: Terry Jane
Scored the highest marks in subject 1
        Roll: 1
        Name: John Kits
Scored the highest marks in subject 2
1. Details of student
2. Average marks
3. Highest marks
4. Exit
Enter your choice :
```

Discussion:

1. This method is useful as it does not require to remember the multiplication tables except for 2.
2. But for negative numbers and floating point numbers this method will not work.
3. Also this program uses iterative approach to fulfill its purpose. Hence it can be problematic for larger numbers. Again, when multiplying with 0, if we set the 0 to the right column and a larger number to the left column then a repetitious 0 will come on the right column.