**Date-**

**Assignment No. :**

**Problem Statement:**

Program in C to store name, roll number and marks of a multiple students, and to perform the following operations on the database :

1. Finding details of a particular student.
2. Calculating the average marks in each subject among all students.
3. Finding the details of the students who scored highest marks in each subject.

**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():

1. Print "Enter number of students: "
2. Input n
3. Repeat step 4 For i = 0 to n-1
4. entry(i)
5. Set i=i+1

[End of For]

1. Repeat through step7 to step 17 while(ch!=4)
2. Print"1. Details of student

2. Average marks

3. Highest marks

4. Exit"

1. Print ”Enter your choice : "
2. Input ch
3. If(ch= 1) Then
4. details()
5. Else If(ch= 2) Then
6. avg()
7. Else If(ch=3) Then
8. highest()
9. Else If(ch!=4)
10. Print "Wrong choice!" [End of If structure]

[End of While loop]

1. Stop

[End of method main()]

Algorithm for method entry(i):

1. Set j=0
2. Print "Enter the details of student "i+1”: “
3. Print ”Enter roll: "
4. Input s[i].roll
5. Print "Enter name: "
6. Input s[i].name
7. Print "Enter marks in subject 1: "
8. Input s[i].marks1
9. Print "Enter marks in subject 2: "
10. Input s[i].marks2
11. Stop

[End of method entry()]

Algorithm for method details():

1. Print "Enter the roll no: "
2. Input r
3. Repeat through step 4 to step 11 For i = 0 to n-1
4. If(s[i].roll = r) Then
5. Print "Details of student:"
6. Print "Name: "s[i].name
7. Print "Marks in:"
8. Print "Subject 1: "s[i].m1
9. Print "Subject 2: "s[i].m2
10. Return

[End of If]

1. Set i=i+1

[End of For loop]

1. If(i = n) Then
2. Print "Record does not exist for roll " r

[End of If structure]

1. Stop

[End of method details()]

Algorithm for method avg():

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

[End of For loop]

1. Set avg1 = t1/n
2. Set avg2 = t2/n
3. Print "Average marks in sub 1: "avg1
4. Print "Average marks in sub 2: "avg2
5. Stop

[End of method avg()]

Algorithm for method highest():

1. Set max1=0
2. Set max2=0
3. Repeat through step 4 to step 10 For i = 0 to n-1
4. If(s[i].m1 > max1)
5. Set max1 = s[i].m1
6. Set t1 = i

[End of If structure]

1. If(s[i].m2 > max2)
2. Set max2 = s[i].m1
3. Set t2 = i

[End of If structure]

1. Set i=i+1

[End of For loop]

1. Print "Details of student(s) who scored the highest marks:"
2. If(t1=t2)
3. Print "Roll: "s[t1].roll "Name: "s[t1].name
4. Print "Scored highest marks in both the subjects"
5. Else
6. Print "Roll: "s[t1].roll "Name: "s[t1].name
7. Print "Scored the highest marks in subject 1"
8. Print "Roll: "s[t2].roll "Name: " s[t2].name
9. Print "Scored the highest marks in subject 2"
10. [End of If-Else structure]
11. 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");

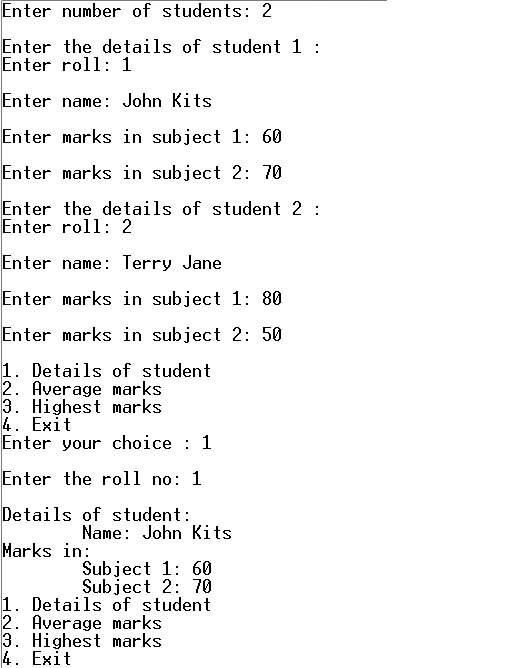
printf("\n\tRoll: %d\n\tName: %s", s[t2].roll, s[t2].name);

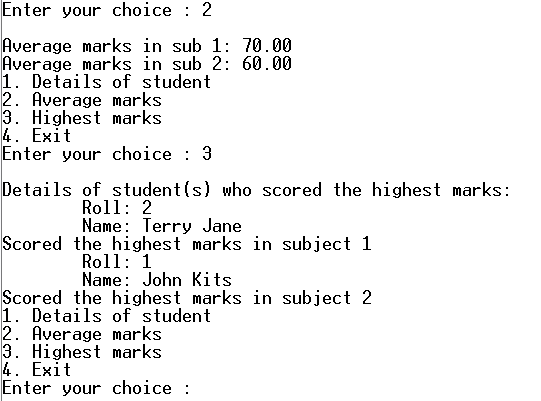
printf("\nScored the highest marks in subject 2");

}

}

**Input & Output:**





**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.