Structure in C

Structure is a user defined data type that allows you to represent a collection of data items of different data type with a single name. ***struct*** is the keyword for this. We may declare a structure ***student*** as follows:

struct student
{
char name{50];
int roll_no;
int age;
};

Here ***struct student*** declares a structure to hold the details of student which consists of three different data fields, namely name, roll_no and age. These are called structure elements. You may take as many data fields as you want. A structure variable ***student*** has been defined here**.** Structure variable declaration is similar to any other basic data type.

Variables can be defined with the structure variable *student*. If there are three students of type *student,* then we should declare as follows

struct student s1, s2, s3;

We can also define structure and declare variables together as ***struct student***

struct student {
char name{50];
int roll_no;
int age;
} s1, s2, s3;

Initialization of structure variables can be done as follows:

struct student s1={"Amit", 10112, 23};
Or (with the dot . operator)
s1.name="Amit";
s1.roll_no=10112;
s1.age=23;

Program to obtain total of three subjects for three students →

```
#include<stdio.h>
struct student
{
char name [50];
int roll_no;
int age;
int sub[3];
};
main()
{
struct student s[3];
int i, j, sum[3];
printf(" Give the detail of three students:\n");
for (i=0; i<3; i++)
{
scanf("%s", s[i].name);
 scanf("%d", s[i].roll_no);
scanf("%d", s[i].age);
for (j=0; j<3; j++)
scanf("%d", s[i].sub[j]);
}
printf("Display of detail of students");
for (i=0; i<3; i++)
{
printf("\n The name of the student %d : %s", i+1,s[i].name);
printf("\n The roll of the student %d is : %d",
i+1,s[i].roll_no);
printf("\n The age of the student %d is: %d", i+1,s[i].age);
for (j=0; j<3; j++)
{
sum[i]=0;
sum[i]=sum[i]+ s[i].sub[j];
}
printf ("The total of student %d is %d", i+1, sum[i]);
}
}
```

Here, typedef keyword may be used in creating a type *stu* (which is of type as struct student**). Then, two structure variables *s1* and s*2* can created by this *stu* type.**

typedef struct student { char name [50]; int roll_no; int age; int sub[3];} stu;

Inside main:
stu s1,s2;

In C, structure can be passed to functions by value (passing actual value as argument)

```
#include<stdio.h>
typedef struct {float r,i;} complex;
complex add (complex a, complex b)
{  complex c;
    c.r = a.r + b.r;
    c.i = a.i + b.i;
   return c;
}
```

```
main( )
{  int n,i; complex x,sum;
    printf("Give n");scanf("%d",&n);
    printf("Give n complex numbers");
    sum.r =0;  sum.i =0;
    for (i=1;i<=n;i++)
    {   scanf("%f%f",&x.r,&x.i);
         sum=add(sum,x);
    }
    printf("Sum=%f+%fi",sum.r,sum.i);
}
```

Assignment

1. Define multiplication function *complex mul (complex a, complex b)* [*(a+bi)(c+di)=(ac-bd)+(ad+bc)*i]
2. Define a structure *point (float a, float b)* and define distance between two points without using function and using function *float distance (point a, point b)*.
3. Define *point mid (point a, point b)*. It finds mid point of A and B.
4. Define function *line equation (point p, point q)* Using it write program which reads two points and finds the equation of line joining them. Let P=(2,3) and Q=(4,7) then line is 2x-y-1=0.