**Data Structures**

Triangle

**Contents:**

* Define a TRIANGLE using typedef with three points.
* Dynamically Allocate memory for n triangles.
* Generate n Triangles by randomly generating x and y coordinates between [10.0, 40.0].
* Writing whole program.

**Defining a triangle using Typedef:**

* First we will create a structure(point) to represent two coordinates X,Y.

//vertex structure  
**typedef struct**{  
 **float** x;  
 **float** y;  
}point;

* Create another structure(TRIANGLE) using the previous structure(point).

//triangle structure   
**typedef struct**{  
 point a;  
 point b;  
 point c;  
 **float** area;  
}TRIANGLE;

* Here a,b,c are points which cointain X and Y coordinates.

**Allocating memory for n triangles.**

* Allocate space for Structure(triangle).
* malloc() for allocating memory dynamically.
* Create a pointer to structure for a triangle and store the address of Dynamically allocated memory in it.

TRIANGLE \*t;  
t=(TRIANGLE\*) malloc((n)\***sizeof**(TRIANGLE));

**Generate n Triangles by randomly generating x and y coordinates between [10.0, 40.0].**

* Make the time as seed for generating random number.
* Generate random no between 10 and 40.
* Assign the random no generated to the point(structure) which contains

x and y(coordinates).

TRIANGLE\* genTriangles(**int** n)  
{  
 srand(time(NULL));  
 TRIANGLE \*t;  
 t=(TRIANGLE\*) malloc((n)\***sizeof**(TRIANGLE));  
 **for**(**int** i=0;i<n;i++) {  
 t[i].a.x=rand()%30+10;  
 t[i].a.y=rand()%30+10;  
  
 t[i].b.x=rand()%30+10;  
 t[i].b.y=rand()%30+10;  
  
 t[i].c.x=rand()%30+10;  
 t[i].c.y=rand()%30+10;  
 }  
 **return** t;  
}

* Here the return type is pointer to structure(TRIANGLE).

**Complete Program:**

#include <stdio.h>  
#include<stdlib.h>  
#include<time.h>  
  
//vertex structure  
**typedef struct**{  
 **float** x;  
 **float** y;  
}point;  
  
//triangle structure  
**typedef struct**{  
 point a;  
 point b;  
 point c;  
 **float** area;  
}TRIANGLE;

TRIANGLE\* genTriangles(**int** n) {  
 srand(time(NULL));  
 TRIANGLE \*t;  
 t=(TRIANGLE\*) malloc((n)\***sizeof**(TRIANGLE));  
 **for**(**int** i=0;i<n;i++) {  
 t[i].a.x=rand()%30+10;  
 t[i].a.y=rand()%30+10;  
  
 t[i].b.x=rand()%30+10;  
 t[i].b.y=rand()%30+10;  
  
 t[i].c.x=rand()%30+10;  
 t[i].c.y=rand()%30+10;  
 }  
 **return** t;  
}  
**int** main(){  
 **int** n=10;  
 TRIANGLE \*t;  
 t=genTriangles(n);  
}