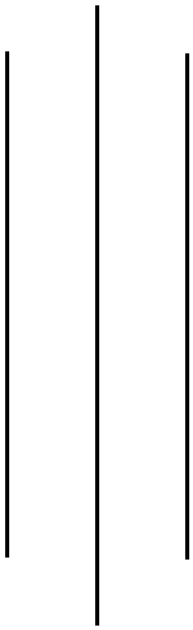
Tribhuvan University

Institute of Science and Technology



LAB SHEET # …….

Submitted by:-

Name:- ……..

Roll no.:- ……

Subject:- ……..

Year:- 1st Year, 1st Semester

Submitted to:-

……

……

……

Submission date:-……..

PROGRAM TO IMPLEMENT TURN TEST (A POINT REFERENCE TO A LINE SEGMENT) AND CHECKING INTERSECTION BETWEEN TWO POINTS

```
#include <iostream>
using namespace std;
class Point{
    public: int x_cor,y_cor;
    void enterPointCoordinate(){
       cout<<"\t\tEnter the X-coordinate: ";
       cin>>x_cor;
       cout<<"\t\tEnter the Y-coordinate: ";
       cin>>y_cor;
    }
    void displayPoint(){
        cout<<"("<<x_cor<<" ,"<<y_cor<<")";
    }
};

class TurnTest{
  public:
  template <class A, class B, class C>
  void enterPointForTurnTest(A& a, B& b, C& c){
      cout<<"\tGiven Line Segment: "<<endl;
      cout<<"\t  Starting Point: "<<endl;
      a.enterPointCoordinate();
      cout<<"\t  End Point: "<<endl;
      b.enterPointCoordinate();
      cout<<"\n\tEnter the third point for which turn test is to be done: "<<endl;
      c.enterPointCoordinate();
      turnTest(a,b,c);
  }
```

```
 template <class A, class B, class C>
 void turnTest(A& a, B& b, C& c){
    double area;
    area = 0.5*(a.x_cor*(b.y_cor-c.y_cor)+b.x_cor*(c.y_cor-a.y_cor)+c.x_cor*(a.y_cor-
    b.y_cor));
    if(area>0){
      cout<<"\n\t\t The point ("<<c.x_cor<<","<<c.y_cor<<") is LEFT to ";
      cout<<"the line segment from: ";
      a.displayPoint();
      cout<<" to ";
      b.displayPoint();
    } else if(area<0){
      cout<<"\n\t\t The point ("<<c.x_cor<<","<<c.y_cor<<") is RIGHT to ";
      cout<<"the line segment from: ";
      a.displayPoint();
      cout<<" to ";
      b.displayPoint();
    } else{
      cout<<"\n\t\t The point ("<<c.x_cor<<","<<c.y_cor<<") is COLLINEAR to ";
      cout<<"the line segment from: ";
      a.displayPoint();
      cout<<" to ";
      b.displayPoint();
    }
 }
};
class Line{
   public: template <class A, class B>
   void line(Point& a, Point& b){
      cout<<"\tFor Starting Point: "<<endl;
      a.enterPointCoordinate();
      cout<<"\tFor End Point: "<<endl;
      b.enterPointCoordinate();
   }
};
```

```
class LineIntersection{
    public:
    template<class A, class B>
    void checkIntersection(A& a, B& b){
        Point p1,p2,p3,p4;
        double p123, p124, p341, p342;
        cout<<"\n\t\t          FIRST LINE";
        cout<<"\n\t\tStart Point:"<<endl;
        p1.enterPointCoordinate();
        cout<<"\n\t\tEnd Point:"<<endl;
        p2.enterPointCoordinate();
        cout<<"\n\t\t          SECOND LINE";
        cout<<"\n\t\tStart Point:"<<endl;
        p3.enterPointCoordinate();
        cout<<"\n\t\tEnd Point:"<<endl;
        p4.enterPointCoordinate();
        p123 = computeArea(p1,p2,p3);
        p124 = computeArea(p1,p2,p4);
        p341 = computeArea(p3,p4,p1);
        p342 = computeArea(p3,p4,p2);
        if (p123 == 0 || p124 == 0 || p341 == 0 || p342 == 0){
            cout<<"\n\t\t-----> The line intersect:Touches <-----"<<endl;
        }
        else if (((p123 > 0 && p124 < 0) && (p341 > 0 && p342 < 0))|| ((p123 > 0 && p124 <
        0) && (p341 < 0 && p342 > 0)) || ((p123 < 0 && p124 > 0) && (p341 < 0 && p342 >
        0))|| ((p123 < 0 && p124 > 0) && (p341 > 0 && p342 < 0))){
            cout<<"\n\t\t-----> Pure Intersection <-----"<<endl;
        }
        else{
            cout<<"\n\t\t-----> line Does not intersect <-----"<<endl;
        }
    }
```

```
template<class A, class B, class C>
   double computeArea(A& a, B& b, C& c){
      return 0.5*(a.x_cor*(b.y_cor-c.y_cor)+b.x_cor*(c.y_cor-a.y_cor)+c.x_cor*(a.y_cor-
      b.y_cor));
   }
};
int main(){
   int choice;
   char cont;
   cout<<"\n\t\t\t **********************************"<<endl;
   cout<<"\t\t\t     POINT LINE CLASSIFICATION "<<endl;
   cout<<"\t\t\t **********************************"<<endl;
   cout<<"\t\t\t   1. Turn Test."<<endl;
   cout<<"\t\t\t   2. Check Intersection between two lines."<<endl;
   cout<<"\n\t\t Enter the choice(1/2): ";
   cin>>choice;
   switch(choice){
      case 1: Point point1, point2, point3;
            TurnTest t1;
            cout<<"\n\t\t\t **********************************"<<endl;
            cout<<"\t\t\t           TURN TEST"<<endl;
            cout<<"\t\t\t **********************************"<<endl;
            t1.enterPointForTurnTest(point1, point2, point3);
         break;
      case 2: LineIntersection li;
            Line l1,l2;
            cout<<"\n\t\t\t **********************************"<<endl;
            cout<<"\t\t\t      Check Intersection"<<endl;
            cout<<"\t\t\t **********************************"<<endl;
            li.checkIntersection(l1,l2);
         break;
      default: cout<<"Invalid choice.\n\tEnter the correct choice number(1/2): ";
   }
   return 0;
                                                }
```

OUTPUT

1. Turn Test

```
                  ****************************************
                    LAB 2: TURN TEST & INTERSECT CHECKING
                  ****************************************
                    1. Turn Test.
                    2. Check Intersection between two lines.

          Enter the choice(1/2): 1

                  **********************************
                             TURN TEST
                  **********************************
  Given Line Segment:
    Starting Point:
          Enter the X-coordinate: 4
          Enter the Y-coordinate: 7
    End Point:
          Enter the X-coordinate: -2
          Enter the Y-coordinate: 1

  Enter the third point for which turn test is to be done:
          Enter the X-coordinate: 2
          Enter the Y-coordinate: 9

           The point (2,9) is RIGHT to the line segment from: (4 ,7) to (-2 ,1)
```

```
                  ****************************************
                    LAB 2: TURN TEST & INTERSECT CHECKING
                  ****************************************
                    1. Turn Test.
                    2. Check Intersection between two lines.

          Enter the choice(1/2): 1

                  **********************************
                             TURN TEST
                  **********************************
Given Line Segment:
  Starting Point:
        Enter the X-coordinate: 2
        Enter the Y-coordinate: 2
  End Point:
        Enter the X-coordinate: 6
        Enter the Y-coordinate: 6

Enter the third point for which turn test is to be done:
        Enter the X-coordinate: 4
        Enter the Y-coordinate: 7

         The point (4,7) is LEFT to the line segment from: (2 ,2) to (6 ,6)
```

2. Intersection Checking

```
        ****************************************
          LAB 2: TURN TEST & INTERSECT CHECKING
        ****************************************
          1. Turn Test.
          2. Check Intersection between two lines.

 Enter the choice(1/2): 2

        **********************************
             Check Intersection
        **********************************
          FIRST LINE
Start Point:
Enter the X-coordinate: 1
Enter the Y-coordinate: 2
End Point:
Enter the X-coordinate: 5
Enter the Y-coordinate: 6

          SECOND LINE
Start Point:
Enter the X-coordinate: 4
Enter the Y-coordinate: 2

End Point:
Enter the X-coordinate: 8
Enter the Y-coordinate: 3

-----> line Does not intersect <-----
```

```
        ****************************************
          LAB 2: TURN TEST & INTERSECT CHECKING
        ****************************************
          1. Turn Test.
          2. Check Intersection between two lines.

 Enter the choice(1/2): 2

        **********************************
             Check Intersection
        **********************************
          FIRST LINE
Start Point:
Enter the X-coordinate: 2
Enter the Y-coordinate: 2
End Point:
Enter the X-coordinate: 6
Enter the Y-coordinate: 6

          SECOND LINE
Start Point:
Enter the X-coordinate: 3
Enter the Y-coordinate: 7

End Point:
Enter the X-coordinate: 7
Enter the Y-coordinate: 1

-----> Pure Intersection <-----
```

```
        ****************************************
          LAB 2: TURN TEST & INTERSECT CHECKING
        ****************************************
          1. Turn Test.
          2. Check Intersection between two lines.

 Enter the choice(1/2): 2

        **********************************
             Check Intersection
        **********************************
          FIRST LINE
Start Point:
Enter the X-coordinate: 2
Enter the Y-coordinate: 5
End Point:
Enter the X-coordinate: 9
Enter the Y-coordinate: 12

          SECOND LINE
Start Point:
Enter the X-coordinate: 4
Enter the Y-coordinate: 7

End Point:
Enter the X-coordinate: 6
Enter the Y-coordinate: 15

-----> The line intersect:Touches <-----
```