****

**Tribhuvan University**

**Institute of Science and Technology**

**LAB SHEET # ……………………….**

**Submitted by:- Submitted to:-**

**Name:- ………………………………………………….. ……………………………………………………………**

**Roll no.:- …………… ……………………………………………………………**

**Subject:- …………………………………….. ……………………………………………………………**

**Year:-** 1st Year, 1st Semester

**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;

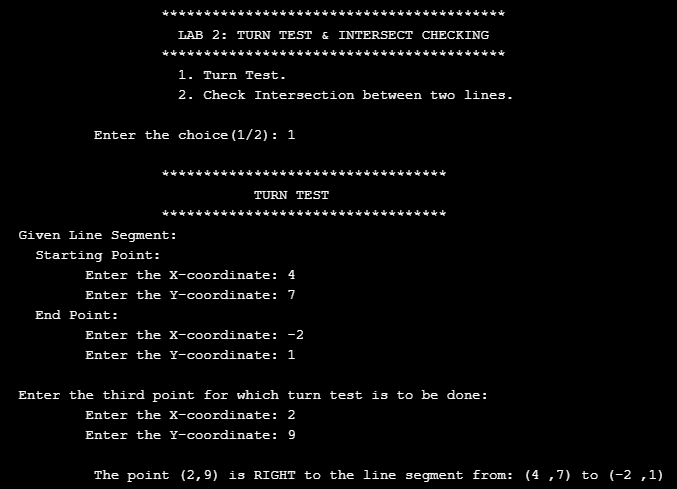
default: cout<<"Invalid choice.\n\tEnter the correct choice number(1/2): ";

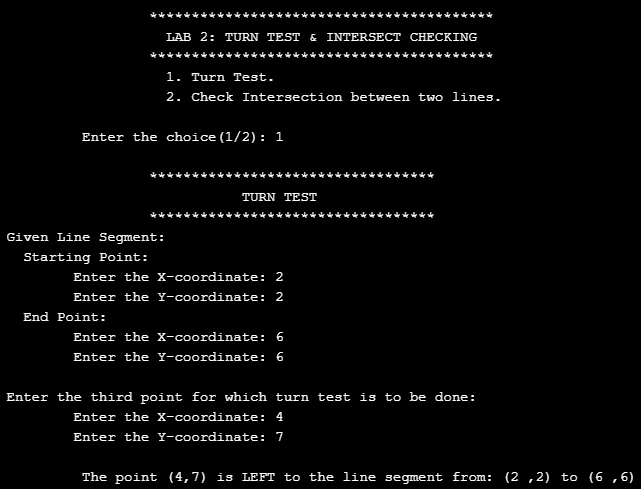
}

return 0;

}**OUTPUT**

1. **Turn Test**





1. **Intersection Checking**

