CS133 Lab 2 – Linear Algebra

Objectives

- Implement basic linear algebra and computational geometry primitives

Detailed Requirements

You are required to implement the set of data types and operations described below as functions and test them on the examples provided.

Data type

Create a Point class which has two double-precision floating point attributes named x and y. Make a synonym to that data type with the name Vector.

Functions

- double CrossProduct(Vector v1, Vector v2)
  Computes the magnitude of the cross product of the two input vectors
- double DotProduct(Vector v1, Vector v2)
  Computes the dot product of the two input vectors
- bool IsCollinear(Point p1, Point p2, Point p3)
  Returns true if and only if the three points are on one straight line
- int Relate(Point p1, Point p2, Point p3)
  Finds the relationship between the straight line (p1, p2) and the point p3. The return value is as follows:
  +ve: Any positive value indicates that p3 is to the right of the ray $\overrightarrow{p_1 p_2}$
  -ve: Any negative value indicates that p3 is to the left of the ray $\overrightarrow{p_1 p_2}$
  0: A zero value indicates that p3 is on the straight line $\overleftrightarrow{p_1 p_2}$
- int Intersection(Point p1, Point p2, Point p3, Point p4, Point &p0)
  Computes the intersection between the two straight lines $\overleftrightarrow{p_1 p_2}$ and $\overleftrightarrow{p_3 p_4}$. The intersection is stored in the output parameter p0. The return value is either 1, 2, or 3 as described below.
  1: Indicates that the two lines are disjoint and parallel. The intersection is empty. p0 is not changed.
  2: Indicates that the two lines are identical. The intersection is a straight line indicates by either $\overleftrightarrow{p_1 p_2}$ or $\overleftrightarrow{p_3 p_4}$. p0 is not changed.
  3: Indicates that the two lines are not parallel. The intersection is a single point returned in the parameter p0.

Examples

- CrossProduct(Vector v1, Vector v2)
    - v1 = (1,-1) and v2 = (2,3) [5]
    - v1 = (2,0) and v2 = (0,2) [4]
    - v1 = (1,3) and v2 = (4,4) [-8]
- DotProduct(Vector v1, Vector v2) , where
    - v1 = (1,-1) and v2 = (2,3) [-1]
    - v1 = (2,0) and v2 = (0,2) [0]
    - v1 = (1,3) and v2 = (4,4) [16]
- IsCollinear(Point p1, Point p2, Point p3) and Relate(Point p1, Point p2, Point p3), where
    - p1= (-5,7), p2=(-4,5), p3=(1,-5) [Collinear]
    - p1= (2,4), p2=(4,6), p3=(6,9) [Not Collinear]
- Relate(Point p1, Point p2, Point p3), where
    - p1= (-5,7), p2=(-4,5), p3=(1,-5) [0]
    - p1= (-30,10), p2=(29,-15), p3=(15,28) [+ve]
    - p1= (5,8), p2=(3,5), p3=(1,3) [-ve]
- Intersection(Point p1, Point p2, Point p3, Point p4, Point &p0)
    - p1= (2,4),p2=(4,8),p3=(1,7),p4=(3,11) [1]
    - p1= (2,4),p2=(4,8),p3=(6,12),p4=(8,16) [2]
    - p1= (1,1),p2=(-5,5),p3=(-9,3),p4=(-4,2), p0=(1,1) [3]