

Lab -

DIP : Design and Analysis of Algorithm

AIM : Implement Geometric Algorithm which check the following

- (1) Are the three given points collinear or not?
- (2) Is the vector POP1 clockwise / counterclockwise with respect to another
vector POP2 ?
- (3) Find the closest-pair of points from the given set of points
- (4) Check INTERSECTION of TWO line Segments

Name : Ambalia Harshit

Department : Computer Engineering (M.Tech sem II)

Roll no : MT001

Date : Jan 2024

Program 01 : Are the three given points collinear or not?

- Code :

```
#include <stdio.h>

struct Point {
    int x, y;
};

int checkCollinear(struct Point p1, struct Point p2, struct Point p3) {
    int slope1 = ( p3.y-p2.y ) * ( p2.x-p1.x );
    int slope2 = ( p2.y-p1.y ) * ( p3.x-p2.x );
    if( slope1==slope2 )
        return 1;
    else
        return 0;
}

int main() {
    struct Point p1, p2, p3;
    scanf("%d %d",&p1.x, &p1.y);
    scanf("%d %d",&p2.x, &p2.y);
    scanf("%d %d",&p3.x, &p3.y);
    if(checkCollinear(p1, p2, p3))
        printf("It is collinear\n");
    else
        printf("It is not collinear\n");
}
```

- Output Screen-shots / Tracing :

```
● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ cd "/home/hr/Documents/Semester_10/Lab_DAA/Lab_11"
● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ gcc -o pr01 && ./pr01
0 0
2 2
3 3
It is collinear
● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$
```

```

● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_10$ gcc ./pr01_m2.c -lm
● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_10$ ./a.out
Text : ccacdaeeba
Pattern : dba
Pattern Not found
○ hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_10$ █

```

Program 02 : Is the vector POP1 clockwise / counterclockwise with respect to another vector POP2 ?

- Code :

```

#include <stdio.h>

struct Point {
    int x, y;
};

int direction_of_p1_wrt_p2(struct Point p0, struct Point p1, struct Point
p2) {
    int tmp1 = p1.x - p0.x;
    int tmp2 = p1.y - p0.y;
    int tmp3 = p2.x - p0.x;
    int tmp4 = p2.y - p0.y;
    int direction = (tmp1*tmp4) - (tmp2*tmp3);
    return direction;
}

int main() {
    struct Point p0, p1, p2;
    scanf("%d %d",&p0.x, &p0.y);
    scanf("%d %d",&p1.x, &p1.y);
    scanf("%d %d",&p2.x, &p2.y);
    int ans = direction_of_p1_wrt_p2(p0, p1, p2);
    if( ans>0 )
        printf("(%d, %d) is counterclockwise to (%d, %d) with base (%d,
%d).\n", p1.x, p1.y, p2.x, p2.y, p0.x, p0.y);
    else if( ans<0 )
        printf("(%d, %d) is clockwise to (%d, %d) with base (%d, %d).\n",
p1.x, p1.y, p2.x, p2.y, p0.x, p0.y);
}

```

```

else
    printf("(%d, %d) and (%d, %d) and (%d, %d) are collinear.\n", p1.x,
p1.y, p2.x, p2.y, p0.x, p0.y);
}

```

- Output Screen-shots / Tracing :

```

● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ cd "/ho
.c -o pr02 && "/home/hr/Documents/Semester_10/Lab_DAA/La
0 0
4 4
1 2
(4, 4) is counterclockwise to (1, 2) with base (0, 0).
○ hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ █

```

```

● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ cd "/ho
.c -o pr02 && "/home/hr/Documents/Semester_10/Lab_DAA/La
0 0
4 4
1 1
(4, 4) and (1, 1) and (0, 0) are collinear.
○ hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11$ █

```

Program 03 : Find the closest-pair of points from the given set of points

- Code :

```
#include <stdio.h>
#include <stdbool.h>
#include <math.h>

typedef struct {
    int x, y;
} Point;

int orientation(Point p, Point q, Point r) {
    int val = (q.y-p.y) * (r.x-q.x) - (q.x-p.x) * (r.y-q.y);
    if( val==0 )
        return 0;
    return( val>0 ) ? 1 : 2;
}

bool on_segment(Point p, Point q, Point r) {
    if( (q.x<=p.x || q.x<=r.x) && (q.x>=p.x || q.x>=r.x) && (q.y<=p.y ||
q.y<=r.y) && (q.y>=p.y || q.y>=r.y) )
        return true;
    return false;
}

bool check_intersection(Point p1, Point q1, Point p2, Point q2) {
    int o1 = orientation(p1, q1, p2);
    int o2 = orientation(p1, q1, q2);
    int o3 = orientation(p2, q2, p1);
    int o4 = orientation(p2, q2, q1);

    if( o1!=o2 && o3!=o4 )
        return true;
    if( o1==0 && on_segment(p1, p2, q1) )
        return true;
    if( o2==0 && on_segment(p1, q2, q1) )
        return true;
    if( o3==0 && on_segment(p2, p1, q2) )
        return true;
```

```

    if( o4==0 && on_segment(p2, q1, q2) )
        return true;
    return false;
}

int main() {
    Point p1 = {1, 1}, q1 = {0, 0};
    Point p2 = {1, 0}, q2 = {0, 1};
    if( check_intersection(p1, q1, p2, q2) )
        printf("Intersects\n");
    else
        printf("Doesn't Intersect\n");
    return 0;
}

```

- Output Screen-shots / Tracing :

41	int main() {
42	Point p1 = {1, 1}, q1 = {0, 0};
43	Point p2 = {1, 0}, q2 = {0, 1};
<div> <div>PROBLEMS</div> <div>OUTPUT</div> <div>DEBUG CONSOLE</div> <div>TERMINAL</div> <div>PORTS</div> </div>	
<pre> ● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ cd "/home/hr/Documents/Semester_10/Lab_DAA/Lab_11" ● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$./pr03.c -o pr03 && "/home/hr/Documents/Semester_10/Lab_DAA/Lab_11/pr03" Intersects ○ hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ </pre>	

42	Point p1 = {1, 1}, q1 = {0, 0};
43	Point p2 = {2, 2}, q2 = {3, 3};
44	if(check_intersection(p1, q1, p2, q2))
<div> <div>PROBLEMS</div> <div>OUTPUT</div> <div>DEBUG CONSOLE</div> <div>TERMINAL</div> <div>PORTS</div> </div>	
<pre> ● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ cd "/home/hr/Documents/Semester_10/Lab_DAA/Lab_11" ● hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$./pr03.c -o pr03 && "/home/hr/Documents/Semester_10/Lab_DAA/Lab_11/pr03" Doesn't Intersect ○ hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ </pre>	

Program 04 : Find the closest-pair of points from the given set of points

○ Code :

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

struct Point {
    int x, y;
};

float distance(struct Point p1, struct Point p2) {
    return sqrt( (p1.x-p2.x)*(p1.x-p2.x) + (p1.y-p2.y)*(p1.y-p2.y) );
}

void closest_pair(struct Point points[], int n) {
    float min_dist = distance(points[0], points[1]);
    int p1 = 0, p2 = 1;

    for( int i=0;i<n-1;i++ ) {
        for( int j=i+1;j<n;j++ ) {
            float dist = distance(points[i], points[j]);
            if( dist<min_dist ) {
                min_dist = dist;
                p1 = i;
                p2 = j;
            }
        }
    }

    printf("Closest pair : (%d, %d) and (%d, %d)\n", points[p1].x,
points[p1].y, points[p2].x, points[p2].y);
    printf("Distance : %f\n", min_dist);
}

int main() {
    struct Point points[] = {{1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10}};
    int n = 5;
    closest_pair(points, n);
    return 0;
}
```

- Output Screen-shots / Tracing :

```
31  int main() {
32      struct Point points[] = {{1, 2}, {3, 4}, {5, 6}, {7, 8}, {9, 10}};
33      int n = 5;
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

- hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ gcc pr04.c -o pr04 -lm
- hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$./pr04
Closest pair : (1, 2) and (3, 4)
Distance : 2.828427
- hr@Edith:~/Documents/Semester_10/Lab_DAA/Lab_11\$ █