Judelle Gaza – CMSC 21 – Lecture 13 Assignment

```c
//Judelle Gaza - CMSC 21 - Lec 13

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

//Structure
struct line{
    struct point{
        float x,y;
    }point1, point2;
    float *midpoint, slope, distance;
};
/*solves the slope using (x,y) points 1 and 2 from the structure */
float solveSlope(struct line linepoint){
    float slope = (linepoint.point1.y - linepoint.point2.y)/(linepoint.point1.x - linepoint.point2.x);
    return slope;
}
/*solves the midpoint using (x,y) points 1 and 2 from the structure and stores into midpoint array*/
float *solveMidpoint(struct line linepoint){
    static float midpoint[2];
    midpoint[0] = (linepoint.point1.x + linepoint.point2.x) / 2;
    midpoint[1] = (linepoint.point1.y + linepoint.point2.y) / 2;
    return midpoint;
}
/*solves the distance of the 2 points using values (x,y) points 1 and 2 from the structure */
float solveDistance(struct line linepoint){
    float x, y;
    x = linepoint.point1.x - linepoint.point2.x;
    y = linepoint.point1.y - linepoint.point2.y;
    float distance = sqrt((x * x) + (y * y));
    return distance;
}
/*finds the value of b in the linear equation and calls the solveSlope() to print into the slope intercept form*/
void getSlopeInterceptForm(struct line linepoint){
    float b = linepoint.point2.y - linepoint.point2.x * solveSlope(linepoint);
    printf("\nSlope Intercept Form: y = %gx + %g", solveSlope(linepoint), b);
}
/* MAIN FUNCTION */
void main(){
    // Declares the points in the structure
    struct line linepoint;

    // Asks and Scans for user input and then stored into the variables in the structure
    printf("Enter (x1,y1) for point 1 [ex. 2 2]: ");
    scanf("%f %f", &linepoint.point1.x, &linepoint.point1.y);
    printf("Enter (x2,y2) for point 2 [ex. 3 3]: ");
    scanf("%f %f", &linepoint.point2.x, &linepoint.point2.y);

    // Prints the output and calls each function
    printf("==================================================\n");
    printf("Slope: %g", solveSlope(linepoint));
    linepoint.midpoint = solveMidpoint(linepoint);
    printf("\nMidpoint: (%g,%g)", *linepoint.midpoint, *(linepoint.midpoint + 1));
    linepoint.distance = solveDistance(linepoint);
    printf("\nDistance between 2 points: %g",linepoint.distance);
    getSlopeInterceptForm(linepoint);
}
```

Example Output:

```
Enter (x1,y1) for point 1 [ex. 2 2]: 4 5
Enter (x2,y2) for point 2 [ex. 3 3]: 8 9
==================================================
Slope: 1
Midpoint: (6,7)
Distance between 2 points: 5.65685
Slope Intercept Form: y = 1x + 1
```