

Mary Nicolette Parcon  
BSCS CMSC 21-1

```
#include <stdio.h>
#include <math.h> /* sqrt(), pow() */

struct line {
    struct point { // (x1, y1), (x2, y2)
        float x;
        float y;
    } point1, point2;
    float slope;
    float midpoint[2]; // since midpoint has 2 points, set as array (midpoint[0],
midpoint[1])
    float distance;
};

// function declarations
void solveSlope(struct line *p);
void solveMidpoint(struct line *p);
void solveDistance(struct line *p);
void getSlopeInterceptForm(struct line *p);

int main(void) {
    struct line line;

    // inputs
    printf("Enter first x and y: ");
    scanf("%f %f", &line.point1.x, &line.point1.y); // (x1, y1)
    printf("Enter second x and y: ");
    scanf("%f %f", &line.point2.x, &line.point2.y); // (x2, y2)

    // use pointers and arrow operators to store values
    solveSlope(&line);
    solveMidpoint(&line);
    solveDistance(&line);

    // outputs
    printf("\nSlope: m = %g", line.slope);
    printf("\nMidpoint = (%g, %g)", line.midpoint[0], line.midpoint[1]);
    printf("\nDistance between two points: d = %g", line.distance);
    getSlopeInterceptForm(&line);
}

void solveSlope(struct line *p) {
```

```

    float slope = (p->point2.y - p->point1.y) / (p->point2.x - p->point1.x); //
(y2 - y1) / (x2 - x1)
    p->slope = slope; // stores slope value to line.slope using arrow operator
}

void solveMidpoint(struct line *p) {
    float x, y;
    x = (p->point1.x + p->point2.x) / 2; // (x1 + x2) / 2
    y = (p->point1.y + p->point2.y) / 2; // (y1 + y2) / 2
    p->midpoint[0] = x; // stores midpoint x value to line.midpoint[0] using
arrow operator
    p->midpoint[1] = y; // stores midpoint y value to line.midpoint[1] using
arrow operator
}

void solveDistance(struct line *p) {
    float x, y, distance;
    x = pow((p->point1.x - p->point2.x), 2); // (x1 - x2)^2
    y = pow((p->point1.y - p->point2.y), 2); // (y1 - y2)^2
    distance = sqrt(x + y); // sqrt(x1 - x2)^2 + (y1 - y2)^2
    p->distance = distance; // stores distance value to line.distance using arrow
operator
}

void getSlopeInterceptForm(struct line *p) {
    float b = p->point1.y - (p->slope * p->point1.x); // b = y - mx, derived from
y = mx + b
    printf("\nSlope-intercept form: y = %gx + %g", p->slope, b); // prints slope-
intercept form using arrow operator to access slope value
}

```

Output:

```

Enter first x and y: 13 5
Enter second x and y: -2 4

Slope: m = 0.0666667
Midpoint = (5.5, 4.5)
Distance between two points: d = 15.0333
Slope-intercept form: y = 0.0666667x + 4.13333
PS C:\Users\mjpar\Documents\UP\SEM 2\CMSC 21-1\Lecture13\Assignments>

```