Computer Engineering Department

TED University

TED UNIVERSITY

CMPE 252 - C Programming, Spring 2021

Lab 1

Part 1 (25 points)

Write a recursive function **`int divisorPower(int num, int d)`** that returns the greatest power of **`d`** where the returned power of **`d`** divides **`num`** evenly (with remainder 0). Assume that both **`num`** and **`d`** are positive integers.

Your task in this part to fill in the missing function definition in skeleton code **`lab1part1.c`**. The remaining part of the code (such as **`main`** function) will stay as it is.

Here are example runs of the program:

```
Enter number> 10
Enter divisor> 2
1 power of 2 divides 10 evenly.
Process returned 0 (0x0)   execution time : 4.334 s
Press any key to continue.
```

```
Enter number> 36
Enter divisor> 3
2 power of 3 divides 36 evenly.
Process returned 0 (0x0)   execution time : 16.136 s
Press any key to continue.
```

```
Enter number> 50
Enter divisor> 7
0 power of 7 divides 50 evenly.
Process returned 0 (0x0)   execution time : 6.669 s
Press any key to continue.
```

Part 2 (75 points)

In this part, you are going to implement the following function in skeleton code **`lab1part2.c`**:

```
void readPointsFindMeans(double *firstMeanXp, double *firstMeanYp, double
*secondMeanXp, double *secondMeanYp);
```

This function is supposed to do the following tasks:

- Read x and y coordinate of the first point using **`scanf`** function.
- Read x and y coordinate of the second point using **`scanf`** function.
- For each of the remaining points, find which point it is closer to (whether it is closer to the first point or the second point) using Euclidean distance measure.
    - Euclidean distance of two points with coordinates (x1, y1) and (x2, y2) can be computed as follows:

$$\sqrt{(x1 - x2)^2 + (y1 - y2)^2}$$

    - If the point has the same distance to both of the points, then the closer point is determined as the first point.
- Compute mean of the points closer to the first point and compute mean of the points closer to the second point.
    - Include the first point while computing mean of the points closer to the first point.
    - Include the second point while computing mean of the points closer to the second point.
- Return computed mean coordinates through the pointer variables listed in the function's formal parameter list.

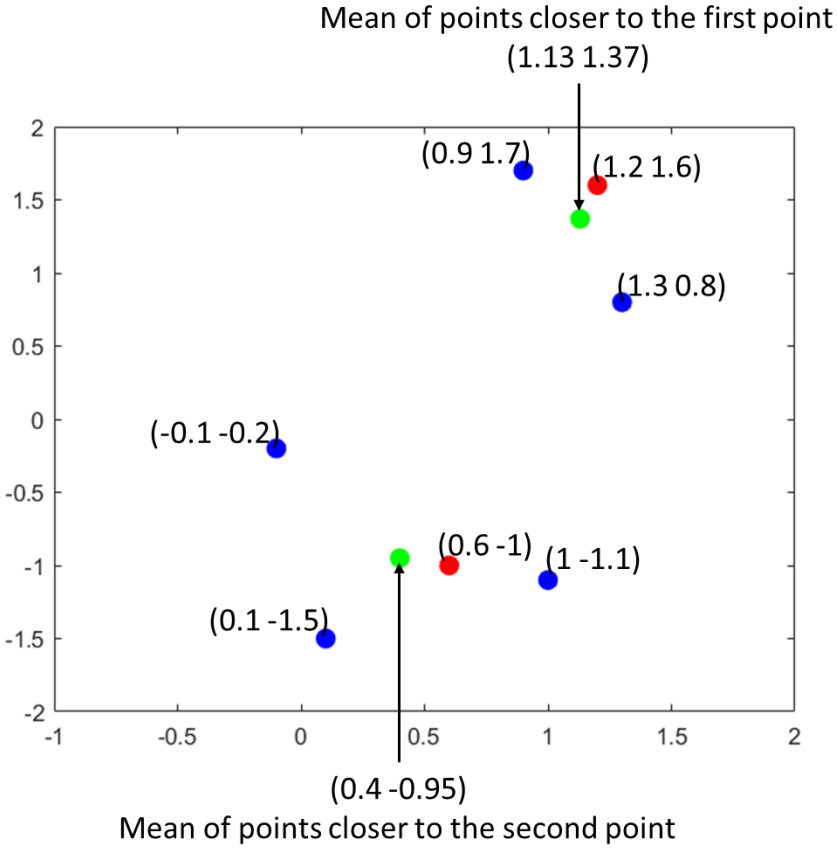
Your task in this part to fill in the missing function definition in skeleton code **`lab1part2.c`**. The remaining part of the code (such as **`main`** function) will stay as it is.

Here is a sample run of the program. Similar to pre-lab, end of input is determined by EOF character.

```
1.2 1.6
0.6 -1
0.9 1.7
1 -1.1
0.1 -1.5
1.3 0.8
-0.1 -0.2
^Z
Mean of points closer to the first point 1.13 1.37
Mean of points closer to the second point 0.40 -0.95

Process returned 0 (0x0)   execution time : 54.530 s
Press any key to continue.
```

The figure below shows the points listed in the run above along with computed mean coordinates. Red points correspond to the first and the second point in the input. Blue points correspond to remaining points in the input. Green points correspond to mean of the points closer to the first point and mean of the points closer to the second point.



Assume that there are at least two points in the input. If there are two points in the input, mean coordinates will be equal to coordinates of the first point and the second point.

Here is another sample run in which there are two points in the input.

```
1.2 -3
-4 -2
^Z
Mean of points closer to the first point 1.20 -3.00
Mean of points closer to the second point -4.00 -2.00

Process returned 0 (0x0)   execution time : 16.017 s
Press any key to continue.
```

Here is another sample run in which there are three points in the input and the third point has the same distance to the first and the second point.

```
1.5 2.5
1.5 -2.5
1 0
^Z
Mean of points closer to the first point 1.25 1.25
Mean of points closer to the second point 1.50 -2.50

Process returned 0 (0x0)   execution time : 41.079 s
Press any key to continue.
```