Section 18 | **PDS Lab Test 1** | **Sep 03, 2019**
Odd set | **Full marks: $100 \times 4 = 400$** | **Time: 3 hours**

Note:

1. Marks will be deducted for inefficient coding, bad structuring of code, bad indentation, lack of important commenting, and deviation from input and output formats as shown in the examples.
2. You cannot use arrays or any library other than `stdio.h` unless mentioned in the question.
3. Name and submit your files as `o1a.c, o1b.c, o1c.c, o1d.c`.
4. In each file you must write your name, roll number, and machine number in the beginning as comment lines.

o1a. **(Triangle area)** Write a function `tri_area` that takes the (x, y)-coordinates of three points A, B, C as arguments, and returns the area of $\triangle ABC$ to `main()`. The six coordinates should be scanned as `float` in `main()` and the area should be printed from `main()`, rounded off to 2nd decimal place.

Note:
You can use `math.h` for the square root function (`sqrtf`).
Area of a triangle is $\sqrt{s(s-a)(s-b)(s-c)}$, where $s = \frac{1}{2}(a+b+c)$.

Examples:

```
Enter (x,y)-coordinates of A: 0 0
Enter (x,y)-coordinates of B: 1 0
Enter (x,y)-coordinates of C: 0 1
Area of ABC =   0.50
```

```
Enter (x,y)-coordinates of A: 1 2
Enter (x,y)-coordinates of B: 3 4
Enter (x,y)-coordinates of C: 5 9
Area of ABC =   3.00
```

```
Enter (x,y)-coordinates of A: 2.1 3.5
Enter (x,y)-coordinates of B: 3.5 7.1
Enter (x,y)-coordinates of C: 7.1 9.3
Area of ABC =   4.94
```

```
Enter (x,y)-coordinates of A: 0 0
Enter (x,y)-coordinates of B: 1 0
Enter (x,y)-coordinates of C: 2 0
Area of ABC =   0.00
```

o1b. **(Large number)** Input are four positive integers, not necessarily distinct. Check whether one among them is larger than thrice of each other. Print "yes" and that number if it exists, otherwise print "no".

Examples:

```
Enter four positive integers: 2 5 3 12
no

Enter four positive integers: 2 12 5 3
no

Enter four positive integers: 2 18 5 3
yes: 18

Enter four positive integers: 3 10 3 3
yes: 10

Enter four positive integers: 3 9 3 3
no
```

o1c. **(New number)** Find the largest digit l in a positive integer n (input). Form a new number m whose digits are l minus the corresponding digits of n. Print l and m.

Examples:

```
Enter n: 25
Largest digit = 5
New number = 30

Enter n: 52
Largest digit = 5
New number = 3

Enter n: 55
Largest digit = 5
New number = 0

Enter n: 152
Largest digit = 5
New number = 403

Enter n: 21394
Largest digit = 9
New number = 78605
```

o1d. **(Largest square)** Given a positive integer n as input, find the largest perfect square s that is at most n. You **cannot** use any multiplication, division, modulo operation, or math library.

Examples:

```
Enter n: 1
s = 1

Enter n: 2
s = 1

Enter n: 4
s = 4

Enter n: 80
s = 64

Enter n: 81
s = 81

Enter n: 82
s = 81

Enter n: 123456
s = 123201
```