Section 18 | **PDS Lab Test 1** | **Sep 03, 2019**
Even 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 `e1a.c, e1b.c, e1c.c, e1d.c`.
4. In each file you must write your name, roll number, and machine number in the beginning as comment lines.

e1a. **(Equilateral triangle perimeter)** Write a function `equi_peri` that takes as arguments the (x, y)-coordinates of the center of a circle C and the (x, y)-coordinates of a point P lying on the circumference of C. It computes the perimeter of the largest equilateral triangle contained in C and returns that value to `main()`. The four coordinates should be scanned as `float` in `main()` and the perimeter should be printed from `main()`, rounded off to 2nd decimal place.

Note:
You can use `math.h` for the square root function (`sqrtf`).

Examples:

```
Enter center coordinates: 1 2
Enter point coordinates:  1 2
Perimeter =   0.00

Enter center coordinates: 1.1 2.1
Enter point coordinates:  2.1 2.1
Perimeter =   5.20
```

```
Enter center coordinates: 0 0
Enter point coordinates:  1 0
Perimeter =   5.20

Enter center coordinates: 5.5 7.1
Enter point coordinates:  2.1 1.2
Perimeter =  35.38
```

e1b. **(Small number)** Input are five positive integers, not necessarily distinct. Check whether there exists one among them such that double that number is less than at least three of them. Print "yes" and that number if it exists, otherwise print "no".

Examples:

```
Enter five positive integers: 3 1 5 2 4
yes: 1

Enter five positive integers: 6 2 5 6 4
yes: 2

Enter five positive integers: 20 5 7 13 11
yes: 5
```

```
Enter five positive integers: 3 2 5 4 4
no

Enter five positive integers: 6 2 3 5 4
no

Enter five positive integers: 20 11 10 22 23
no
```

e1c. **(New number)** Reverse a positive integer p (input) to get a new integer q, and find all the common divisors of p and q that are greater than 1. Print "none" if there is none.

Examples:

```
Enter p: 6
Reverse number = 6
Common divisors: 2, 3, 6.

Enter p: 23
Reverse number = 32
Common divisors: none.

Enter p: 24
Reverse number = 42
Common divisors: 2, 3, 6.

Enter p: 920
Reverse number = 29
Common divisors: none.

Enter p: 314586
Reverse number = 685413
Common divisors: 3, 9.

Enter p: 314587
Reverse number = 785413
Common divisors: none.

Enter p: 868
Reverse number = 868
Common divisors: 2, 4, 7, 14, 28, 31, 62, 124, 217, 434, 868.
```

e1d. **(Smallest power)** Given a positive integer n as input, find the smallest integer p such that 5^p is not less than n, and print that value of 5^p. You **cannot** use any multiplication, division, modulo operation, or math library.

Examples:

```
Enter n: 1
min 5^p= 1

Enter n: 4
min 5^p= 5

Enter n: 5
min 5^p= 5

Enter n: 6
min 5^p= 25

Enter n: 24
min 5^p= 25

Enter n: 25
min 5^p= 25

Enter n: 26
min 5^p= 125

Enter n: 1000
min 5^p= 3125
```