

Python Review: Conditional Processing, Counted Repetition

Answer the following questions.

1. Explain the difference between the two blocks of code below.

```
if x == 4:                if x == 4:
    print("A")            print("A")
if x > 0:                elif x > 0:
    print("B")            print("B")
```

2. Explain the purpose of each of the following commands, associated with `for` loops.
 - a. `break`
 - b. `continue`
 - c. `else`

Write programs that accomplish each task. Use proper conventions for variable names, input prompts, output statements, and program structure. At this point in time, assume that the user will enter the correct data type, but not necessarily the desired value.

3. Obtain two integers, x and y , from the user and display all integer values between x and y inclusive. For example, when $x = 5$ and $y = 9$, your program should display 5, 6, 7, 8, 9.
4. Generate ten random integers between 1 and 10 (inclusive) and count the number of odd and even values. For example, the sequence 3, 5, 6, 2, 3, 2, 1, 9, 4, 1 has six odd and four even values.
5. Generate three random positive integers, and determine if they can form the sides of a triangle. Classify the triangle as right, acute or obtuse. For example, side lengths of 3, 5 and 9 do not form a triangle, while side lengths of 4, 5 and 6 form an acute triangle.
6. Read a positive integer, n , between 1 and 10, and display an $n \times n$ grid representing the multiplication table up to n . For example, $n = 3$ results in the following:

1	2	3
2	4	6
3	6	9

7. A **Heronian triangle** is a triangle whose side lengths (a , b , c) and area (A) are integers. The simplest Heronian triangle is the right-angled triangle with side lengths 3, 4 and 5 units, and an area of 6 square units. Verify that there are 288 Heronian triangles whose side lengths do not exceed 100 units, and whose area does not exceed 2 000 square units. It may be useful to use Heron's formula for the area of a triangle: $A = \sqrt{s(s-a)(s-b)(s-c)}$, where $s = \frac{a+b+c}{2}$. The `floor` or `ceil` functions in the `math` module might also be useful.