

## PRACTICE EXERCISES - LAB 2 (Loops)

1. Here is a description of Euclid's algorithm to print the GCD (greatest common divisor) of positive integers  $x$  and  $y$ :

**Repeat until**  $y$  is 0

**Set**  $r$  to be the remainder when  $x$  is divided by  $y$ .

**Copy** the value of  $y$  into  $x$ .

**Copy** the value of  $r$  into  $y$ .

**Print**  $x$

Write and run a Python program that asks the user for two positive integers, and uses Euclid's algorithm to find and print the GCD of those integers.

**Hint:** Use the `%` operator to find remainders. For example, the Python expression `42 % 30` will evaluate to 12.

2. Write and run a Python program that asks the user to enter 8 integers (one at a time), and then prints out how many of those integers were even numbers. For example, if the user entered 19, 6, 9, 20, 13, 7, 6, and 1, then your program should print out 3, since 3 of those numbers were even.

**Hint:** You don't need to remember the 8 numbers. You only need to remember how many of the numbers entered so far have been even.

**Hint:** Use the remainder (`%`) operator to test if an integer is even.

3. Here is an algorithm for printing the maximum value in a sequence:

```
Initialize max_so_far
For each value in the sequence
    If value is greater than max_so_far
        Copy value into max_so_far
print max_so_far
```

Write and run a Python program that asks the user how many values they would like to enter. Then ask the user for that many positive numbers, and use the algorithm above to determine the highest of those numbers. Your program should terminate by printing that highest number. (What should be the initial value of *max\_so\_far*?)

**Challenge:** Modify your program so that it does not ask the user how many values will be entered. Instead, the user will simply enter numbers, one at a time. Eventually, they will enter the number 0 to indicate that there are no more numbers in the sequence. (What is the weakness of this plan?) Then, your program should print out the highest of those numbers.

4. Write a program that asks the user for a positive integer, and prints PRIME if the number is a prime number, and otherwise prints COMPOSITE.

**Challenge:** Write a program that asks the user for a number  $n$ , and prints all prime numbers from 2 to  $n$ .

**Challenge:** Now modify that program so that it instead prints the first  $n$  prime numbers.

5. Write a program that asks the user how many days are in a particular month, and what day of the week the month begins on (0 for Monday, 1 for Tuesday, etc), and then prints a calendar for that month. For example, here is the output for a 30-day month that begins on day 4 (Thursday):

S	M	T	W	T	F	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	