

Name: \_\_\_\_\_ **SOLUTION** \_\_\_\_\_ CM: \_\_\_\_\_ Section: \_\_\_\_\_ Grade: \_\_\_\_\_ of 10

1. Here is a correct implementation of a function that returns the sum of the cubes of the integers from  $m$  to  $n$ , inclusive:

```
def sum_cubes(m, n):
    """ Returns the sum of the cubes of the integers from m to n, inclusive. """
    total = 0
    for k in range(m, n + 1):
        total = total + (k ** 3)
    return total
```

Write an alternative implementation that uses a WHILE loop instead of a FOR loop.

|  |  |
|--|--|
| <pre>def sum_cubes(m, n):     total = 0     k = m     while True:         if k &gt; n:             break         total = total + (k ** 3)         k = k + 1     return total</pre> | <p><i>Note: Many alternatives are possible, including:</i></p> <pre>total = 0 k = m while k &lt;= n:     total = total + (k ** 3)     k = k + 1 return total</pre> |
|--|--|

2. Which of the above two implementations is more easily/quickly understood (hence better)? (circle your choice)

The implementation using a FOR loop

The implementation using a WHILE loop

Why? *It is shorter and the RANGE statement summarizes the behavior of k succinctly and clearly.*

3. Consider the following problem:

Implement a function that returns the sum of the first  $N$  integers after (and including)  $M$ . For example, if  $M$  is **10** and  $N$  is **6**, this function would return **10 + 11 + 12 + 13 + 14 + 15**, which is 75.

For the above problem, which is a better choice? (circle your choice)

An implementation using a FOR loop

An implementation using a WHILE loop

4. Consider the following problem:

Implement a function that returns the sum of the first  $N$  integers after (and including)  $M$  **that are prime**. For example, if  $M$  is **10** and  $N$  is **6**, this function would return **11 + 13 + 17 + 19 + 23 + 29**, which is 112.

This problem CANNOT be solved by using a FOR loop. Explain why not.

*A FOR loop requires a RANGE (or the equivalent), which requires knowing (when the loop starts) how many iterations the loop will run. The loop here has to go up by 1 (or 2) each time. We cannot say in advance how many times that increment by 1 (or 2) is needed to reach the Nth prime.*

5. Write a statement that prompts for and inputs an integer from the Console. (See *m1r* for how to do this problem.) `m = int(input("Enter an integer: "))`