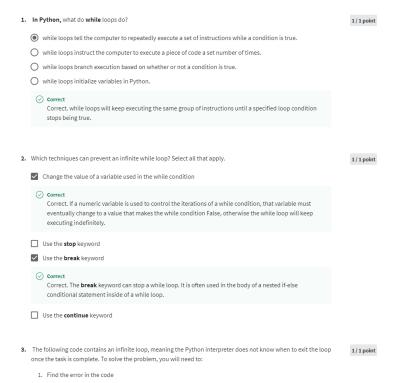
Congratulations! You passed!

Grade received 80% To pass 80% or higher

Go to next item



Hint: Try running your function with the number 0 as the input and observe the result.

2. Fix the while loop so there is an exit condition

Note that Coursera's code blocks will time out after 5 seconds of running an infinite loop. If you get this timeout error message, it means the infinite loop has not been fixed.

Correct
 Correct. You fixed a tricky error that was hard to find and the function now behaves correctly.

4. Fill in the blanks to complete the while loop so that it returns the sum of all the divisors of a number, without including the number itself. As a reminder, a divisor is a number that divides into another without a remainder. To do this, you will need to:

1/1 point

- Initialize the "divisor" and "total" variables with starting values
- 2. Complete the while loop condition
- 3. Increment the "divisor" variable inside the while loop
- 4. Complete the return statement

```
# Fill in the blanks so that the while loop continues to run while the
# "divisor" variable is less than the "number" parameter.

def sum_divisors(number):
# Initialize the appropriate variables
divisor = 1

total = 0

# Avoid dividing by 0 and negative numbers
# in the while loop by exiting the function
# if "number" is less than one
finumber < 1:
```

5. Fill in the blanks to complete the function, which should output a multiplication table. The function accepts a variable "number" through its parameters. This "number" variable should be multiplied by the numbers 1 through 5, and printed in a format similar to "1x6=6" ("number" x "multiplier" = "result"). The code should also limit the "result" to not exceed 25. To satisfy these conditions, you will need to:

0/1 point

- 1. Initialize the "multiplier" variable with the starting value
- 2. Complete the while loop condition
- 3. Add an exit point for the loop
- 4. Increment the "multiplier" variable inside the while loop

Not quite. First, the variable "multiplier" should be initialized to equal the number 1. The while loop should run "while multiplier = 5:". If the "result" is greater than 25, use the "break" keyword to break out of the loop. Finally, increment the "multiplier" variable by 1. To review the materials on this topic, please see the "More While Loop Examples", "Why Initializing Variables Matters", and "Infinite Loops and How to Break Them" videos.