[Jahangirnagar University](https://www.coursera.org/programs/jahangirnagar-university-na-qyc9d)

[Skip to Main Content](https://www.coursera.org/learn/python-crash-course/quiz/mDS8z/practice-quiz-while-loops/attempt#main)

* 

MD. TAZEL HOSSAN

[Crash Course on Python](https://www.coursera.org/learn/python-crash-course/home/welcome)

[Week 3](https://www.coursera.org/learn/python-crash-course/home/week/3)

Practice Quiz: While Loops

[Prev](https://www.coursera.org/learn/python-crash-course/supplement/k4CY9/infinite-loops-and-code-blocks)

[Next](https://www.coursera.org/learn/python-crash-course/lecture/C9zN4/what-is-a-for-loop)

**PRACTICE QUIZ • 25 MIN**

**Practice Quiz: While Loops**

**Submit your assignment**

Try again

**Receive grade**

**TO PASS**80% or higher

**Grade**

80%

View Feedback

We keep your highest score

Practice Quiz: While Loops

Practice Quiz • 25 min

**Practice Quiz: While Loops**

**TOTAL POINTS 5**

1.Question 1

What are while loops in Python?

1 point



While loops let the computer execute a set of instructions while a condition is true.



While loops instruct the computer to execute a piece of code a set number of times.



While loops let us branch execution on whether or not a condition is true.



While loops are how we initialize variables in Python.

2.Question 2

Fill in the blanks to make the print\_prime\_factors function print all the prime factors of a number. A prime factor is a number that is prime and divides another without a remainder.

1 point

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

def print\_prime\_factors(number):

  # Start with two, which is the first prime

  factor = 2

  # Keep going until the factor is larger than the number

  while factor <= number:

    # Check if factor is a divisor of number

    if number % factor == 0:

      # If it is, print it and divide the original number

      print(factor)

      number = number / factor

    else:

      # If it's not, increment the factor by one

      factor+=1

  return "Done"

print\_prime\_factors(100)

# Should print 2,2,5,5

# DO NOT DELETE THIS COMMENT





RunReset

3.Question 3

The following code can lead to an infinite loop. Fix the code so that it can finish successfully for all numbers.

Note: Try running your function with the number 0 as the input, and see what you get!

1 point

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

def is\_power\_of\_two(n):

  # Check if the number can be divided by two without a remainder

  if n!=0:

    while n % 2 == 0:

      n = n / 2

  # If after dividing by two the number is 1, it's a power of two

  if n == 1:

    return True

  return False

print(is\_power\_of\_two(0)) # Should be False

print(is\_power\_of\_two(1)) # Should be True

print(is\_power\_of\_two(8)) # Should be True

print(is\_power\_of\_two(9)) # Should be False





RunReset

4.Question 4

Fill in the empty function so that it returns the sum of all the divisors of a number, without including it. A divisor is a number that divides into another without a remainder.

1 point

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

def sum\_divisors(n):

  sum = 0

  if n==0:

    return sum

  else:

    temp = 1

    x = n

    while n>0:

      r = int(n / temp)

      if r!=0 and n%r==0:

        sum+=r

        temp+=1

      n = r

    return sum-x

  # Return the sum of all divisors of n, not including n

print(sum\_divisors(0))

# 0

print(sum\_divisors(3)) # Should sum of 1

# 1

print(sum\_divisors(36)) # Should sum of 1+2+3+4+6+9+12+18

# 55

print(sum\_divisors(102)) # Should be sum of 2+3+6+17+34+51

# 114





RunReset

5.Question 5

The multiplication\_table function prints the results of a number passed to it multiplied by 1 through 5. An additional requirement is that the result is not to exceed 25, which is done with the break statement. Fill in the blanks to complete the function to satisfy these conditions.

1 point

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

def multiplication\_table(number):

    # Initialize the starting point of the multiplication table

    multiplier = 1

    # Only want to loop through 5

    while multiplier <= 5:

        result =number\*multiplier

        # What is the additional condition to exit out of the loop?

        if result>25 :

            break

        print(str(number) + "x" + str(multiplier) + "=" + str(result))

        # Increment the variable for the loop

        multiplier+= 1

multiplication\_table(3)

# Should print: 3x1=3 3x2=6 3x3=9 3x4=12 3x5=15

multiplication\_table(5)

# Should print: 5x1=5 5x2=10 5x3=15 5x4=20 5x5=25

multiplication\_table(8)

# Should print: 8x1=8 8x2=16 8x3=24





RunReset



I, **MD. TAZEL HOSSAN**, understand that submitting work that isn’t my own may result in permanent failure of this course or deactivation of my Coursera account. Learn more about Coursera’s Honor Code