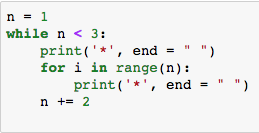
1. x = list(range(2, 6, 4)), x = \_\_\_\_\_\_\_\_\_\_\_

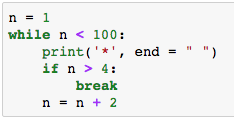
x = list(range(range(5, 0, -1)), x = \_\_\_\_\_\_\_\_\_\_\_

x = len(range(1, 5, 2), x = \_\_\_\_\_\_\_\_\_\_\_

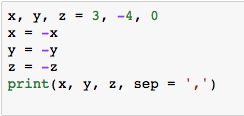
1. How many asterisks will the following Python code print? \_\_\_\_\_\_\_\_\_



1. How many asterisks will the following Python code print? \_\_\_\_\_\_\_\_\_



1. The following Python code prints \_\_\_\_\_\_\_\_\_\_

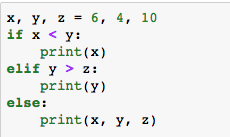


1. The following Python code prints \_\_\_\_\_\_\_\_\_\_

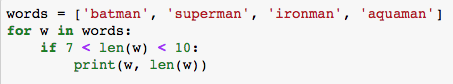
../../../../Desktop/Screen%20Shot%202018-02-21%20at%202.29.27%

1. The following Python code prints (2 answers) \_\_\_\_\_\_\_\_\_\_

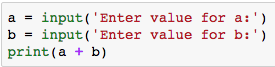
\_\_\_\_\_\_\_\_\_\_



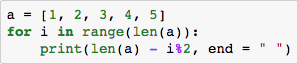
1. The following Python code prints \_\_\_\_\_\_\_\_\_\_



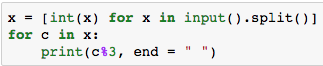
1. The following Python code prints \_\_\_\_\_\_\_\_\_\_ assuming 2 and 22 are entered for *a* and *b*, respectively.



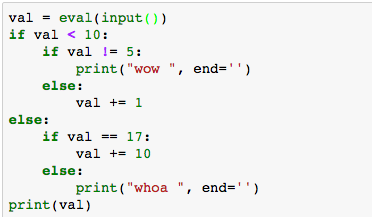
1. The following Python code prints \_\_\_\_\_\_\_\_\_\_



1. The following Python code prints \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ if 1 2 3 4 5 6 7 8 9 are entered by a user.



1. Consider the following Python code. What are the outputs if 5, 9 and 17 are entered?

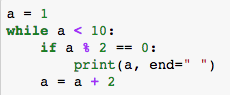


Input = 5, output = \_\_\_\_\_\_\_\_\_\_\_\_

Input = 9, output = \_\_\_\_\_\_\_\_\_\_\_\_

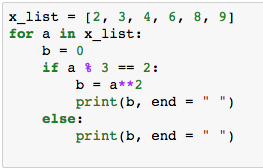
Input = 17, output = \_\_\_\_\_\_\_\_\_\_\_\_

1. The following Python code is developed aiming to print 2 4 6 8 on the screen. However, it does not produce the intended result. What Python statement should be added to fix the issue?



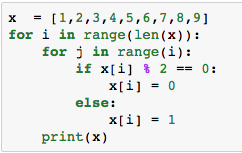
Expected output is: 2 4 6 8

1. (2 marks) Trace the following Python code and fill in the blank.



|  |  |  |
| --- | --- | --- |
| a | if a % 3 == 2  (True or False) | print(b, end = " ") |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. (3 mark) How many times will list *x* be printed on the screen? What are the values in list *x* for each print?



How many times? \_\_\_\_\_\_\_\_\_\_\_\_

For each time, print(x) = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_