In the box below, write a boolean expression that evaluates to true if and only if the variable x is positive (greater than zero) and even. Assume that x is an integer.

You will want to use remainder (the % operator) to test for parity. If you are unsure of how to use it, run the following lines in a python interpreter to experiment with its usage.

10 % 2
10 % 3
10 % 4
11 % 2
11 % 3
11 % 4

(x > 0) and (x % 2 == 0)

Needs manual grading
Answer:
Feedback:
(x > 0) and (x % 2 == 0)

Consider these variables:

name = "Alice"
age = 20
weight = 120.0

Using the variables above, select the statement that will print the exact output:

Alice,20,120.000000

- print "%%s,%%d,%%f" % (name, age, weight)
- print name, ",", age, "", weight
- print "%%s,%%d,%%f" % name % age % weight
Consider a file, input.txt, which stores the names, ages, and weights of various people:

Alice
20
120.0
Bob
17
130.5
Freddie
21
190.6

In the box below, write the printed output that results from the execution of the following code.

```python
input = open("input.txt", "r")
age_sum = 0
people_count = 3
for line in range(people_count):
    name = input.readline()
age = int(input.readline())
weight = float(input.readline())
age_sum = age_sum + age
print float(age_sum) / people_count
input.close()
```

```
19.3333333333
```
Consider a file, input.csv, containing this text:

Alice,20,120.0
Bob,17,130.5
Freddie,21,190.6

In the box below, write the contents of output.txt that results from the execution of the following code.

```python
input = open("input.csv", "r")
output = open("output.txt", "w")

line_number = 0
for line in input:
    if line_number % 2 == 0:
        output.write(line)
    line_number = line_number + 1

input.close()
output.close()
```

Alice,20,120.0
Freddie,21,190.6

Needs manual grading

Answer:

Feedback:
Alice,20,120.0
Freddie,21,190.6