Tutorial 2

Q1. What is the output of the following Python program?

```
value = 6
if value % 2 == 0:
   print("first", value)
elif value % 3 == 0:
   print("second", value)

while value <= 9:
   value = value + 1
   if value == 8:
      continue
   else:
      pass
   print ("third", value)
else:
   print ("fourth", value)

print("fifth", value)
```

Answer	Discussion

first 6	`value % 2 == 0 is True, enter True, although`
	`value % 3 == 0 is True, it is skipped`

third 7	6. + 1 =7, display “third 7” pass has no effect (does nothing) but helps in **indicating** an empty statement/ suite/ block. When 7 + 1, value becomes 8, • The **continue** statement continues with the next iteration of the loop. • Skip some portion of the **while** suite we are executing and have control flow back to the beginning of the **while** loop. • Exit early from this iteration of the loop (not the loop itself), and keep executing the **while** loop.
third 9	
third 10	
fourth 10	While-else • It is entered after the **while** loop’s Boolean expression becomes False. • This entry occurs even when the expression is initially False and the **while** loop has never run. • A handy way to perform some final tasks when the loop ends normally.
fifth 10	Statement after while loop

Q2. The following program calculates the number of input strings with letter ‘a’, and end the program when the input string is “####”. Here is an expected sample run:

Sample :

```
enter a string (enter #### to stop): apple
enter a string (enter #### to stop): banana
enter a string (enter #### to stop): strawberry
enter a string (enter #### to stop): book
enter a string (enter #### to stop): ####
3 strings with letter 'a'
```

```
while True:
    str = input("enter a string: ")
    for letter in str:
      if letter == 'a':
          break
      count +=1

    print(count , "strings with letter 'a'")
```

There are some errors in the above program. Please indicate where the errors are and how to correct them.

```
count=0
while True:
    str_sentinal  = input("enter a string (enter #### to stop):  ")
    if str_sentinal =="####":
        break
    for letter in str_sentinal :
      if letter == 'a':
          count +=1
          break


print(count , "strings with letter 'a'")
```

```python
count = 0
str_sentinal = input("enter a string (enter #### to stop): ")
while str_sentinal != "####":
    for letter in str_sentinal:
      if letter == 'a':
          count +=1
          break
    str_sentinal = input("enter a string (enter #### to stop): ")
print(count , "strings with letter 'a'")
```

Q3 There is a sequence called the Fibonacci sequence. The first two numbers in the Fibonacci sequence are both 1, and the third number (as well as the remaining numbers in the sequence) is the sum of the previous two.

$$1,\ 1,\ 2,\ 3,\ 5,\ 8,\ 13,\ 21,\ 34,\ 55,\ 89,\ 144,\ \ldots$$

The sequence *F_n* of Fibonacci numbers is defined by the recurrence relation:

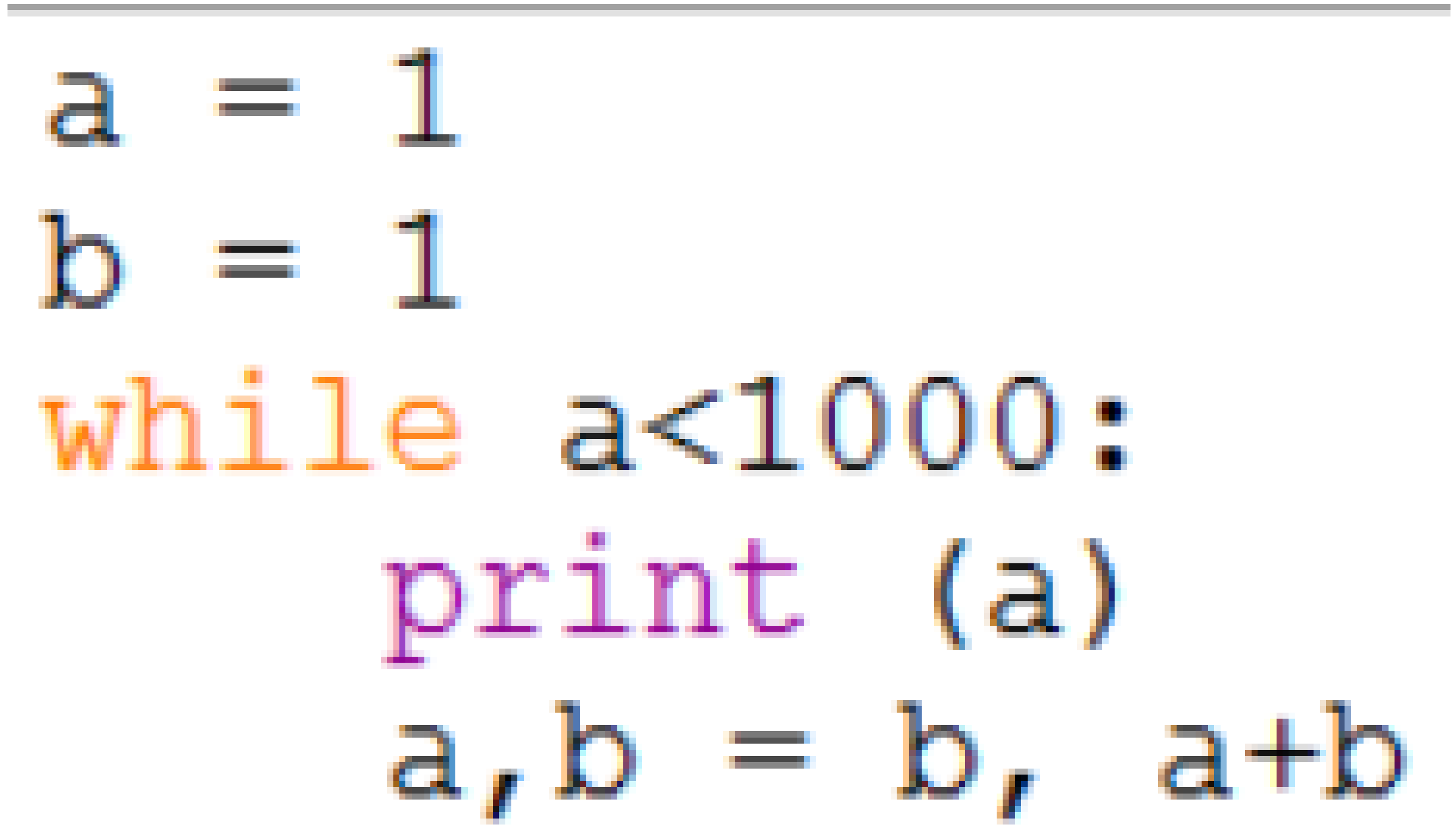
$$F_n = F_{n-1} + F_{n-2},$$

with seed values

$$F_1 = 1,\ F_2 = 1$$

Write a simple Python program to generate this sequence before 1000. Note: use multiple assignments with a simple while loop to compute.

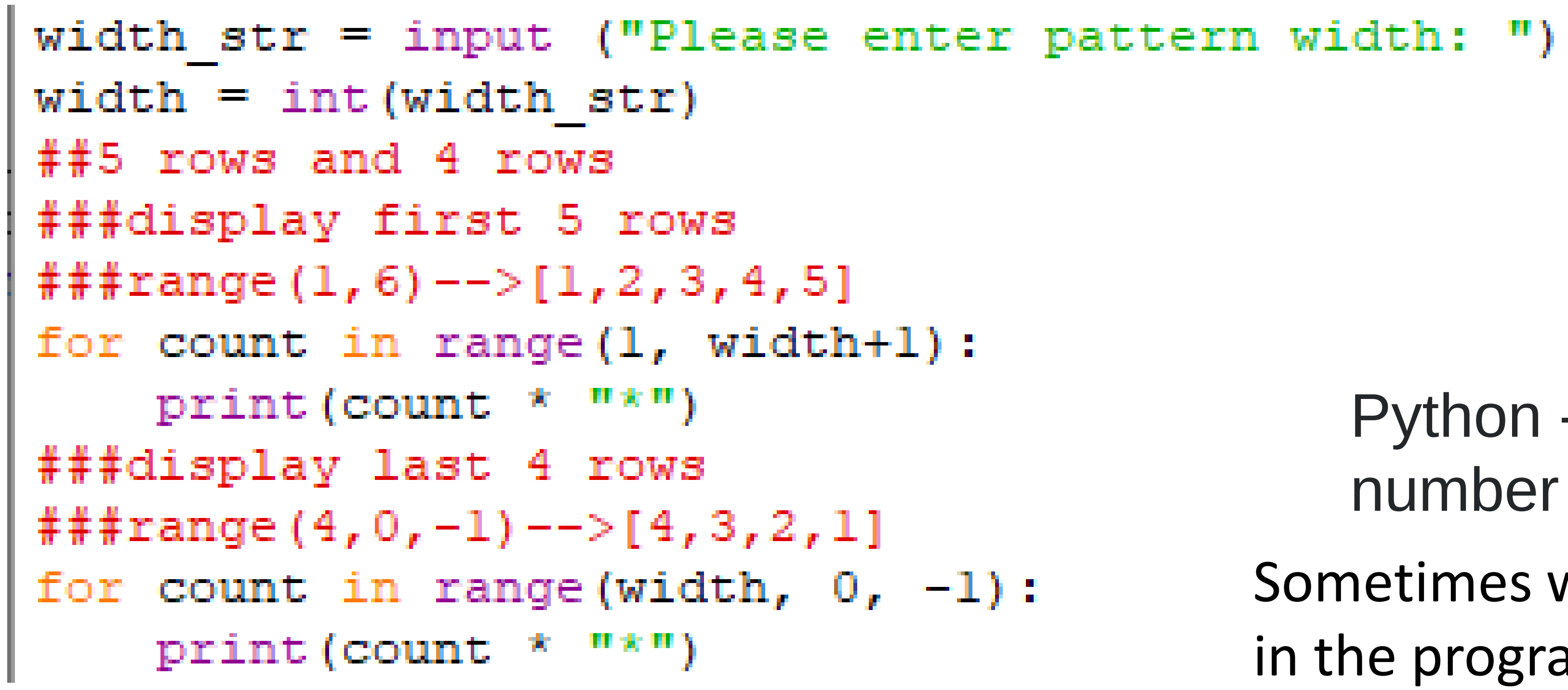
Check this for your own interest: https://en.wikipedia.org/wiki/Fibonacci_number



```
a = 1
b = 1
while a<1000:
    print (a)
    a,b = b, a+b
```

Q4. Write a Python program that reads an integer from the user, which is the width of the pattern below, and then prints out the pattern. Suggestion: use nested **for** loops. Hint: **print("*",end="")**.

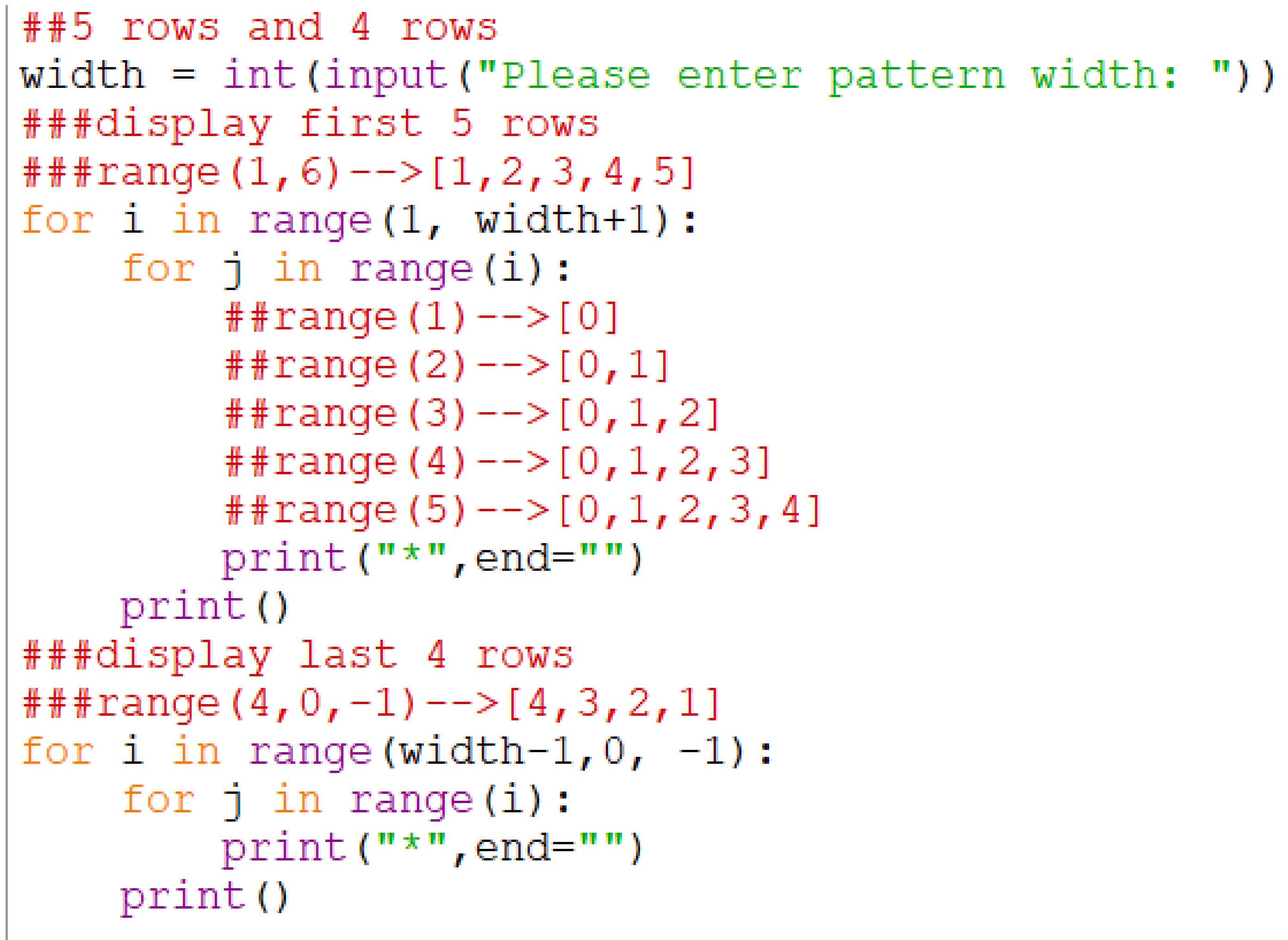
```
Please enter pattern width: 5

*
**
***
****
*****
****
***
**
*
```



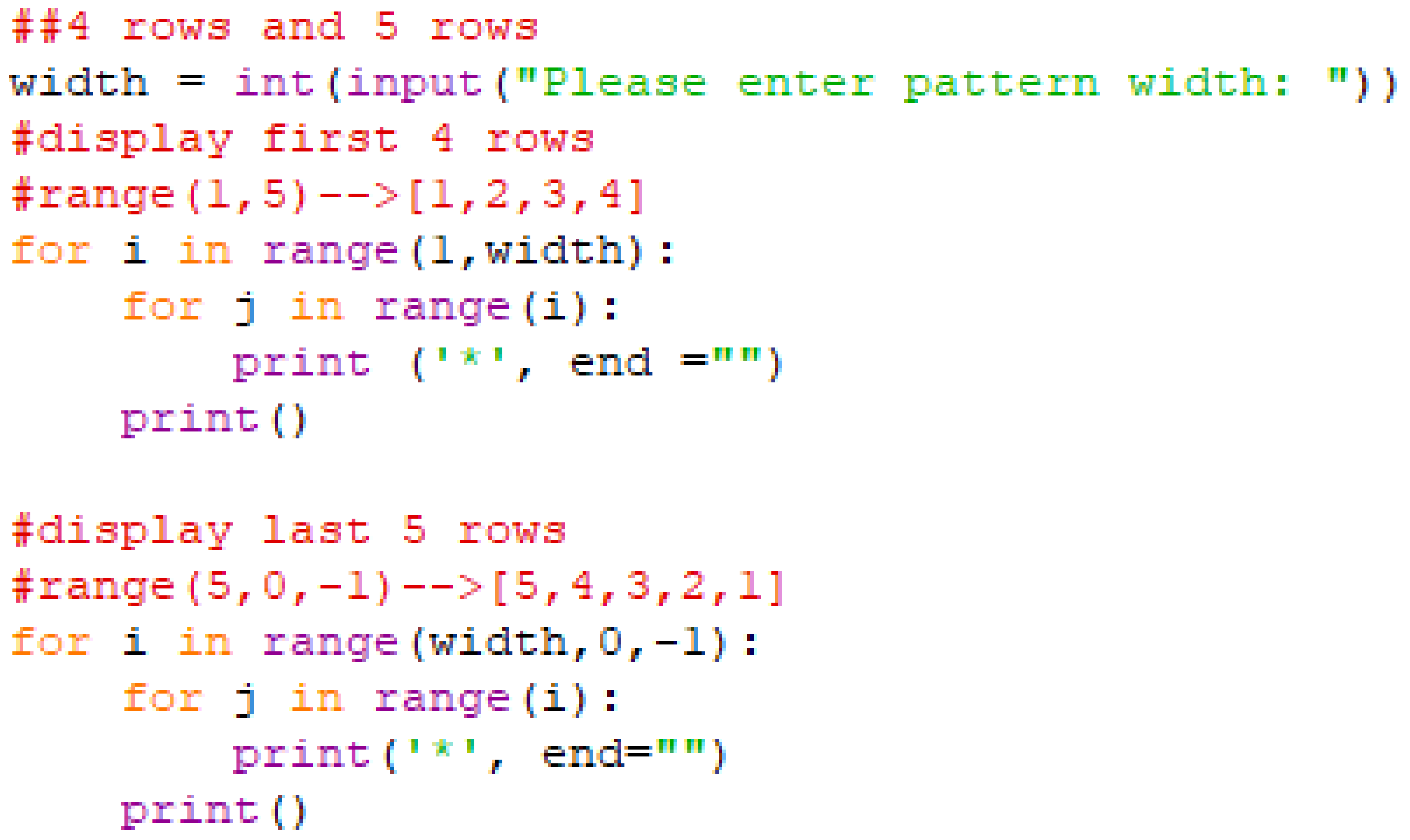
```
width_str = input ("Please enter pattern width: ")
width = int(width_str)
##5 rows and 4 rows
###display first 5 rows
###range(1,6)-->[1,2,3,4,5]
for count in range(1, width+1):
    print(count * "*")
###display last 4 rows
###range(4,0,-1)-->[4,3,2,1]
for count in range(width, 0, -1):
    print(count * "*")
```

Python - Print a string a certain number of times [duplicate]

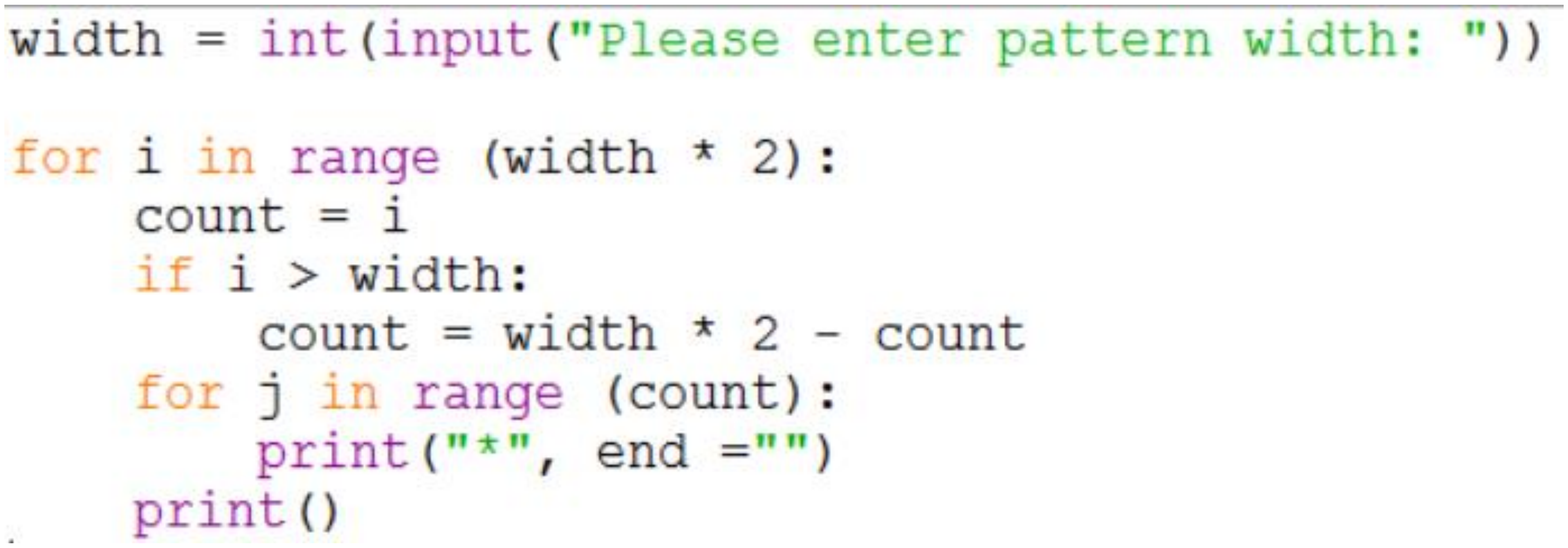
Sometimes we need to repeat the string in the program, and we can do this easily by using the repetition operator in Python.
Repetition operator is denoted by a '*' symbol and is useful for repeating strings to a certain length.



```
##5 rows and 4 rows
width = int(input("Please enter pattern width: "))
###display first 5 rows
###range(1,6)-->[1,2,3,4,5]
for i in range(1, width+1):
    for j in range(i):
        ##range(1)-->[0]
        ##range(2)-->[0,1]
        ##range(3)-->[0,1,2]
        ##range(4)-->[0,1,2,3]
        ##range(5)-->[0,1,2,3,4]
        print("*",end="")
    print()
###display last 4 rows
###range(4,0,-1)-->[4,3,2,1]
for i in range(width-1,0, -1):
    for j in range(i):
        print("*",end="")
    print()
```

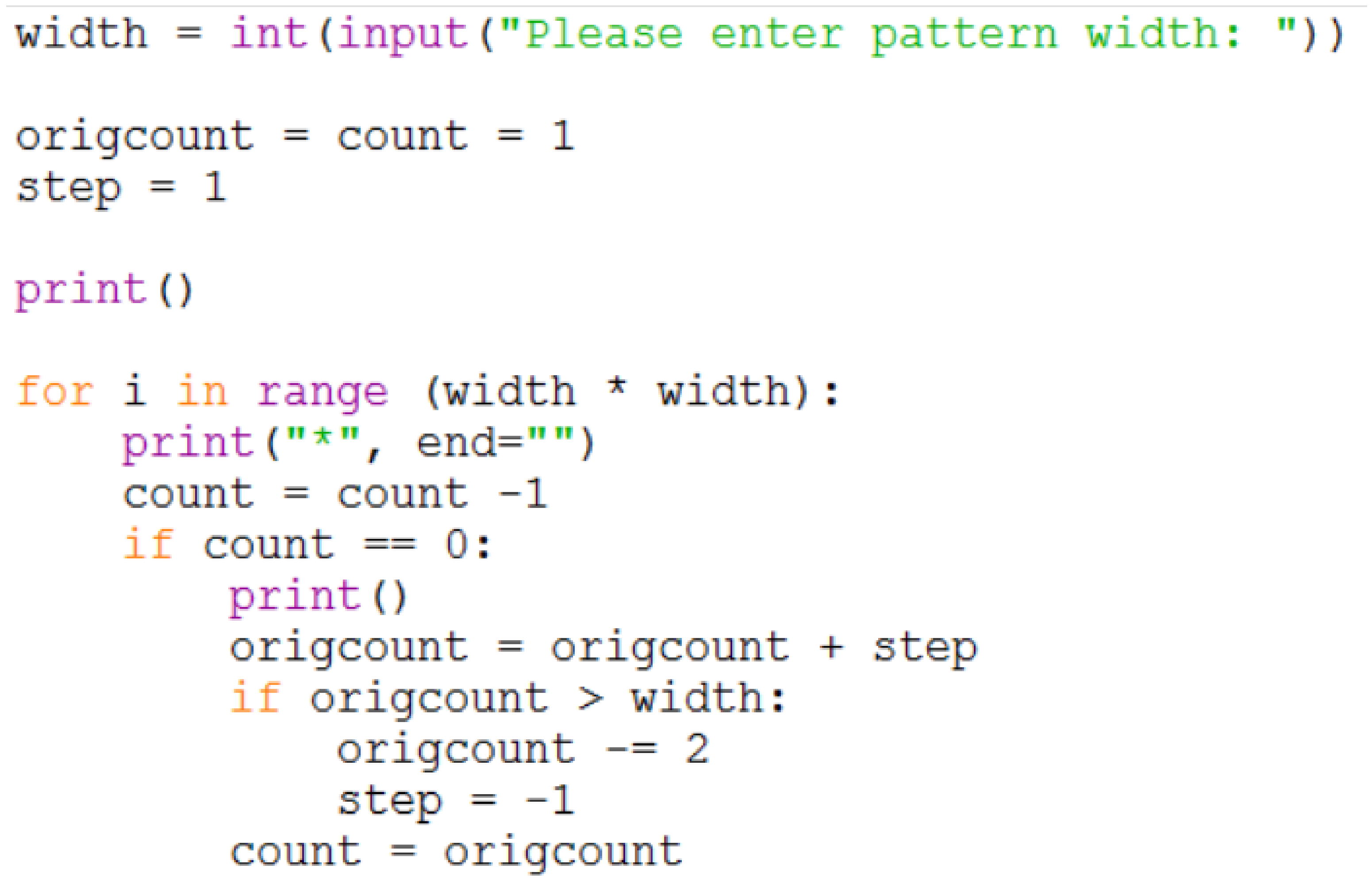


```
##4 rows and 5 rows
width = int(input("Please enter pattern width: "))
#display first 4 rows
#range(1,5)-->[1,2,3,4]
for i in range(1,width):
    for j in range(i):
        print ('*', end ="")
    print()

#display last 5 rows
#range(5,0,-1)-->[5,4,3,2,1]
for i in range(width,0,-1):
    for j in range(i):
        print('*', end="")
    print()
```



```python
width = int(input("Please enter pattern width: "))

for i in range (width * 2):
    count = i
    if i > width:
        count = width * 2 - count
    for j in range (count):
        print("*", end ="")
    print()
```

```python
width=int(input('Please enter pattern width = '))

count=0
for i in range(2*width+1):
    count+=1
    if count<=width+1:
        print('*'*i)
        j=i
    else:
        j-=1
        print('*'*j)
```



```python
width = int(input("Please enter pattern width: "))

origcount = count = 1
step = 1

print()

for i in range (width * width):
    print("*", end="")
    count = count -1
    if count == 0:
        print()
        origcount = origcount + step
        if origcount > width:
            origcount -= 2
            step = -1
        count = origcount
```