**Discussion 1**

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)

**Discussion 2**

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.

**Discussion 3**

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.



The sequence *Fn* of Fibonacci numbers is defined by the [recurrence relation](https://en.wikipedia.org/wiki/Recurrence_relation):



with [seed values](https://en.wikipedia.org/wiki/Seed_value)

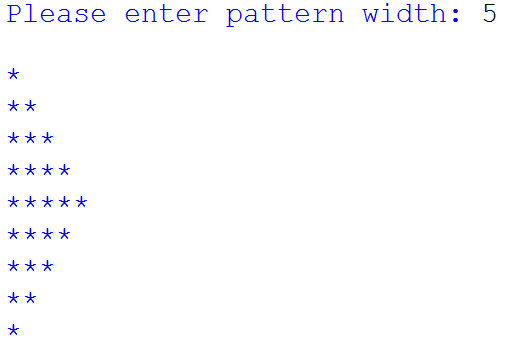


Write a simple Python program to generate this sequence before 1000. Note: use multiple assignments with a simple while loop to compute.

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

**Discussion 4**

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="")**.

****

Further discussion: Is it possible to use **for** only twice? Or even once? (of course no **while**)