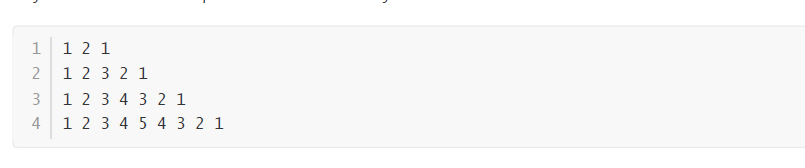
Week 1 - PPA 8 to 13

**Question-8**

**Statement**

Print the following pattern. There is exactly one space between any two consecutive numbers on any line. There are no spaces at the end of any line.



# Answer

# print('1 2 1')

# print('1 2 3 2 1')

# print('1 2 3 4 3 2 1')

# print('1 2 3 4 5 4 3 2 1')

# 

# 

# **Question-9**

# **Statement**

# 

# Answer

# rollno = input()

# branch = rollno[:2]

# print(branch == 'DS')

# 

# **Question-10**

# **Statement**

# 

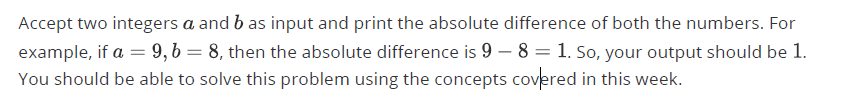
Answer

number = input()

print(number[:4] == "TN07" or number[4:] == "TN07")

**Question-11**

**Statement**



# **Answer**

# a = int(input())

# b = int(input())

# square = (b - a) \*\* 2

# root = square \*\* 0.5

# diff = int(root)

print(diff)

# 

**Question-12**

**Statement**

You are given a string and two non-negative integers as input. The two integers specify the start

and end indices of a substring in the given string. Create a new string by replicating the substring a minimum number of times so that the resulting string is longer than the input string.

The input parameters are the string, start index of the substring and the end index of substring

(endpoints inclusive) each on a different line.

**Answer**

word = input()

start = int(input())

end = int(input())

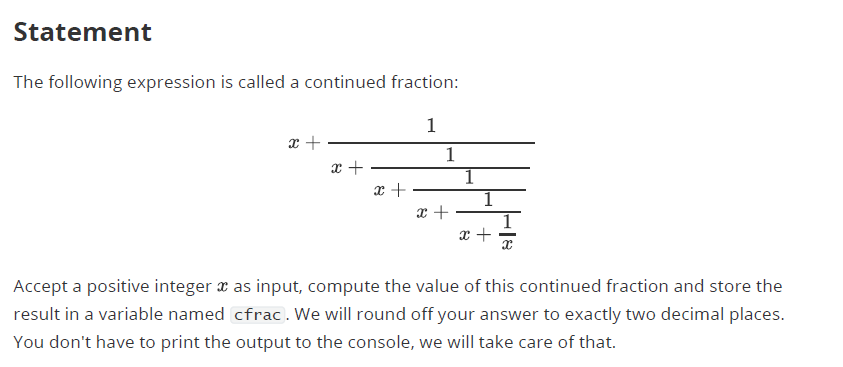
substring = word[start : end+1]

min\_rep = len(word) // (end - start + 1) + 1

substring = substring \* min\_rep

print(substring)

**Question-13**



**Answer**

x = int(input())

l0 = x + 1 / x

l1 = x + 1 / l0

l2 = x + 1 / l1

l3 = x + 1 / l2

cfrac = x + 1 / l3