Started on	Friday, 29 March 2024, 6:58 PM
State	Finished
Completed on	Friday, 29 March 2024, 7:09 PM
Time taken	10 mins 59 secs
Marks	5.00/5.00
Grade	**50.00** out of 50.00 (**100**%)
Name	DWIJESH SREERAM S 2022-CSD-A

Write a program to find the count of ALL digits in a given number N. The number will be passed to the program as an input of type int.

Assumption: The input number will be a positive integer number>= 1 and<= 25000.

For e.g.

If the given number is 292, the function should return 3 because there are 3 digits in this number

If the given number is 1015, the function should return 4 because there are 4 digits in this number
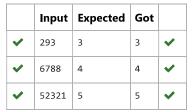
For example:

InputResult

292   3

1015 4

For example:

Input	Result
293	3

Answer: (penalty regime: 0 %)

```python
def count(n):
    num=str(n)
    return len(num)

n=int(input())
d=count(n)
print(d)
```

	Input	Expected	Got	
✔	293	3	3	✔
✔	6788	4	4	✔
✔	52321	5	5	✔

Passed all tests! ✔

Correct
Marks for this submission: 1.00/1.00.

You are choreographing a circus show with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive direction.

•The first kangaroo starts at position x1 and moves at a speed v1 meters per jump.

•The second kangaroo starts at position x2   and moves at a speed of v2 meters per jump and x2 > x1

•You have to figure out to get both kangaroos at the same position at the same time as part of the show before k jumps. If it is possible, return YES, otherwise return NO.

Input Format:

x1-position of kangaroo1

v1-Speed of kangaroo1

x2-position of kangaroo2

v2-Speed of kangaroo2

k-jumps

Output Format:

Both kangaroos are at the same position within k jumps, YES, otherwise NO.

For example:

Input	Result
0 3 4 2 6	YES

Answer: (penalty regime: 0 %)

```
1  def kangaroo_meet(x1, v1, x2, v2, k):
2      # If kangaroo 1 is slower than kangaroo 2 or both have same speed, they will never meet
3      if v1 <= v2:
4          return "NO"
5
6      # If the initial positions are such that they can't meet even after first jump
7      if (x2 - x1) % (v1 - v2) != 0:
8          return "NO"
9
10     # Calculate the number of jumps required for kangaroo 1 to catch up with kangaroo 2
11     jumps_needed = (x2 - x1) // (v1 - v2)
12
13     # If number of jumps required is within the limit
14     if 0 < jumps_needed <= k:
15         return "YES"
16     else:
17         return "NO"
18
19 # Input
20 x1=int(input())
21 v1=int(input())
22 x2=int(input())
```

```
23  v2=int(input())
24  k=int(input())
25
26  # Output
27  print(kangaroo_meet(x1, v1, x2, v2, k))
28
```

	Input	Expected	Got	
✔	0 3 4 2 6	YES	YES	✔
✔	0 3 2 4 8	NO	NO	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Write a program that reads a positive integer, n, from the user and then displays the sum of all of the integers from 1 to n.

Sample Input

10

Sample Output

The sum of the first 10 positive integers is 55.0

For example:

Input	Result
10	The sum of the first 10 positive integers is 55.0

Answer: (penalty regime: 0 %)

```python
1  def sum_of_integers(n):
2      # Sum = (n * (n + 1)) / 2
3      sum_val = (n * (n + 1)) / 2
4      return sum_val
5
6  n = int(input())
7  result = sum_of_integers(n)
8  print("The sum of the first", n, "positive integers is", result)
9
```

	Input	Expected	Got	
✔	10	The sum of the first 10 positive integers is 55.0	The sum of the first 10 positive integers is 55.0	✔
✔	20	The sum of the first 20 positive integers is 210.0	The sum of the first 20 positive integers is 210.0	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

A number is stable if each digit occur the same number of times.i.e, the frequency of each digit in the number is the same. For e.g. 2277,4004,11,23,583835,1010 are examples for stable numbers.

Similarly, a number is unstable if the frequency of each digit in the number is NOT same.

Sample Input:

2277

Sample Output:

Stable Number

Sample Input 2:

121

Sample Output 2:

Unstable Number

For example:

Input	Result
2277	Stable Number

Answer: (penalty regime: 0 %)

```python
def is_stable(number):
    # Convert number to string for easy iteration
    num_str = str(number)

    # Count the frequency of each digit using a dictionary
    digit_frequency = {}
    for digit in num_str:
        digit_frequency[digit] = digit_frequency.get(digit, 0) + 1

    # Check if all frequencies are the same
    frequencies = list(digit_frequency.values())
    return all(freq == frequencies[0] for freq in frequencies)

# Input
number = input()

# Check stability
if is_stable(number):
    print("Stable Number")
else:
    print("Unstable Number")
```

	Input	Expected	Got	
✔	9988	Stable Number	Stable Number	✔
✔	2277	Stable Number	Stable Number	✔
✔	1233	Unstable Number	Unstable Number	✔

Passed all tests! ✔

Question **5**

Correct

Mark 1.00 out of 1.00

Determine the factors of a number (i.e., all positive integer values that evenly divide into a number).

For example:

Input	Result
20	1 2 4 5 10 20

Answer: (penalty regime: 0 %)

```python
 1  def find_factors(number):
 2      factors = []
 3      for i in range(1, number + 1):
 4          if number % i == 0:
 5              factors.append(i)
 6      return factors
 7
 8  number = int(input())
 9  factors = find_factors(number)
10  for factor in factors:
11      print(factor, end=" ")
12
```

	Input	Expected	Got	
✔	20	1 2 4 5 10 20	1 2 4 5 10 20	✔
✔	5	1 5	1 5	✔
✔	13	1 13	1 13	✔

Passed all tests! ✔

Jump to...