

Q2)

A)

INPUT

```
def CalculateDigits(number):  
    if number < 0:  
        raise ValueError("The input needs to be a whole number that is greater than zero.")  
  
    return len(str(number))  
  
# Verify the function  
num = 618  
result = CalculateDigits(num)  
print(f"The total count of digits within {num} is: {result}")
```

OUTPUT

```
Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]  
Type "copyright", "credits" or "license" for more information.  
  
IPython 7.22.0 -- An enhanced Interactive Python.  
  
In [1]: runfile('C:/Users/rm6197r/.spyder-py3/untitled0.py', wdir='C:/Users/rm6197r/.spyder-py3')  
The total count of digits within 618 is: 3  
  
In [2]: |
```

CODE WRITTEN:

```
def CalculateDigits(number):  
    if number < 0:  
        raise ValueError("The input needs to be a whole number that is greater than zero.")  
  
    return len(str(number))  
  
# Verify the function  
num = 618  
result = CalculateDigits(num)  
print(f"The total count of digits within {num} is: {result}")
```

EXPLANATION:

Definition of the CalculateDigits function:

The digit count of an integer is determined by this function.

First, if the given number is a positive whole number (greater than or equal to zero), it is checked.

The function generates a ValueError with a message informing the user that the input must be a positive whole number if the value is negative.

If the number is correct, str(number) is used to convert it to a string, and len() is used to determine the length of that string, which is equal to the integer's digit count.

This count of digits is then returned by the function.

Calling the function CalculateDigits(num):

Using the integer 618 as an example, this line of code invokes the CalculateDigits function.

The variable result contains the outcome of this function call.

Outputting the Outcome:

Lastly, it outputs the function call's total count of digits as well as the original number, 618.

To sum up, this code defines a function that counts the digits in a positive integer and prints the result after determining how many digits there are in the integer 618.

B)

INPUT

```

1  def hasMoreOrEqualDigits(a, b):
2      if a < 0 or b < 0:
3          raise ValueError("The numbers given need to be integers that are greater than zero.")
4
5      return len(str(a)) >= len(str(b))
6
7  # Experimenting with the function using integers generated randomly
8  import random
9
10 # Creating integers at random within a specific range [10, 9999]
11 integers_a = [123, 23, 1234]
12 integers_b = [234, 352, 568]
13
14 # Evaluating the function using the given values.
15 predicted_outcome = [True, False, True]
16 for i in range(len(integers_a)):
17     result = hasMoreOrEqualDigits(integers_a[i], integers_b[i])
18     expected = predicted_outcome[i]
19     print(f"a({integers_a[i]}), b({integers_b[i]}) , return value({result}) - Expected: {expected}")
20
21

```

OUTPUT

```

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.22.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/rm6197r/.spyder-py3/untitled1.py', wdir='C:/Users/rm6197r/.spyder-py3')
a(123), b(234) , return value(True) - Expected: True
a(23), b(352) , return value(False) - Expected: False
a(1234), b(568) , return value(True) - Expected: True

In [2]:

```

CODE WRITTEN:

```
def hasMoreOrEqualDigits(a, b):
```

```
    if a < 0 or b < 0:
```

```
        raise ValueError("The numbers given need to be integers that are greater than zero.")
```

```

return len(str(a)) >= len(str(b))

# Experimenting with the function using integers generated randomly
import random

# Creating integers at random within a specific range [10, 9999]
integers_a = [123, 23, 1234]
integers_b = [234, 352, 568]

# Evaluating the function using the given values.
predicted_outcome = [True, False, True]
for i in range(len(integers_a)):
    result = hasMoreOrEqualDigits(integers_a[i], integers_b[i])
    expected = predicted_outcome[i]
    print(f"a({integers_a[i]}), b({integers_b[i]}) , return value({result}) - Expected: {expected}")

```

EXPLANATION

(hasMoreOrEqualDigits) Function Definition:

When two integers, a and b, are compared, the function hasMoreOrEqualDigits(a, b) determines whether a has more or equal to b in terms of digits.

It initially determines whether a or b are negative. It reports a ValueError and indicates that both values must be positive integers if one of them is negative.

In the event that both numbers are valid, the number of digits in each is calculated by using str() to convert the numbers to strings. The lengths of both strings are then compared using len() to ascertain whether a has more or equal to b's number of digits.

If a has greater or equal to b's number of digits, the function returns True; if not, it returns False.

Function Execution: The code proceeds to test the `hasMoreOrEqualDigits` function using sets of integers (`integers_a` and `integers_b`). It assigns expected outcomes to each test case using the `predicted_outcome` list. **Test Cases:** The for loop iterates through the test cases by comparing each pair of integers from `integers_a` and `integers_b`. For each pair, it calculates the result using the `hasMoreOrEqualDigits` function. It compares this result with the expected outcome (`predicted_outcome`) and prints the values of a, b, the return value, and the expected result for each test case. In summary, this code tests the `hasMoreOrEqualDigits` function by comparing the number of digits in pairs of integers and verifies if the function returns the expected results for those comparisons.