WEEK 4:

1.A Number is said to be Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself. Write a program to print number is Disarium or not.

Input Format: Single Integer Input from stdin.

Output Format: Yes or No.

Example 1:

Input: 175

Output: Yes

Explanation 1^1+7^2+5^3 = 175

Example 2:

Input: 123

Output: No

**Program:**

num=input()

n = len(num)

res = 0

for i, digit in enumerate(num):

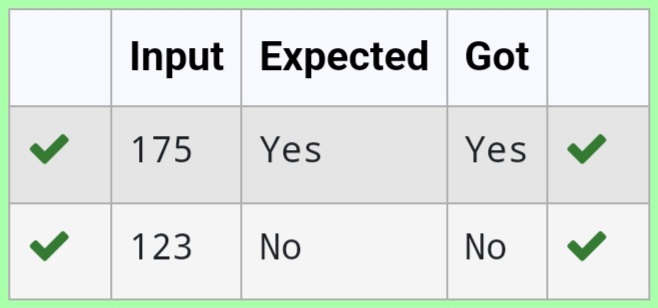
res += int(digit)\*\*(i+1)

if res == int(num):

print(“Yes”)

else:

print(“No”)

**Output**:

2.Write a program to find the count of non- repeated 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.

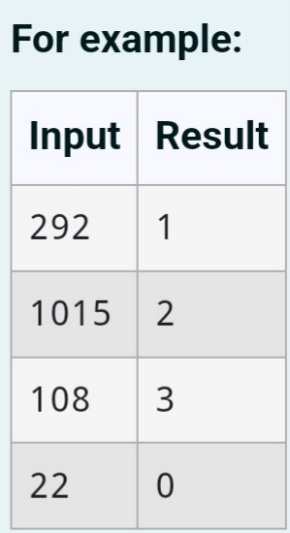
Some examples are as below.

If the given number is 292, the program should return 1 because there is only 1 non-repeated digit ‘9’ in this number

If the given number is 1015, the program should return 2 because there are 2 non- repeated digits in this number, ‘0’, and ‘5’.

If the given number is 108, the program should return 3 because there are 3 non—repeated digits in this number, ‘1’, ‘0’, and ‘8’.

If the given number is 22, the function should return 0 because there are NO non—repeated digits in this number.2.



**Program:**

num = input()

c = 0

for digit in set(num):

if num.count(digit) == 1:

c+=1

print(c)

**Output:**

****

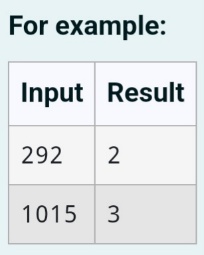
3. Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

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

For e.g.

If the given number is 292, the program should return 2 because there are only 2 unique digits ‘2’ and ‘9’ in this number

If the given number is 1015, the program should return 3 because there are 3 unique digits in this number, ‘1’, ‘0’, and ‘5’.



**Program:**

n = input()

c = len(set(n))

print(c)

**Output:**

****

4. Write a program to return the nth number in the fibonacci series.

The value of N will be passed to the program as input.



**Program:**

n = int(input())

a, b = 0, 1

for x in range(n-1):

a, b = b, a + b

print(a)

**Output:**

****

5. Given an integer N, check whether N the given number can be made a perfect square after adding to it.

Input Format: Single integer input.

Output Format: Yes or No.

Example

**Input**: 24

**Output**: Yes

Example

**Input**: 26

**Output**: No

**Program:**

import math

num = int(input())

Sqrt\_num = math.isqrt(num + 1)

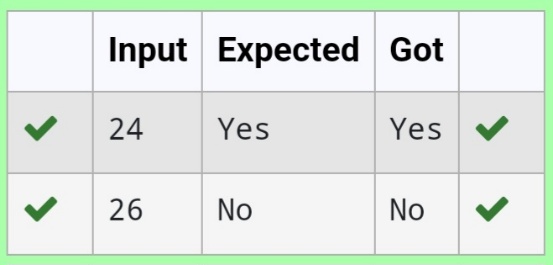
if num + 1 == sqrt\_num \* sqrt\_num:

Print(“Yes”)

else:

Print(“No”)

**Output:**

****

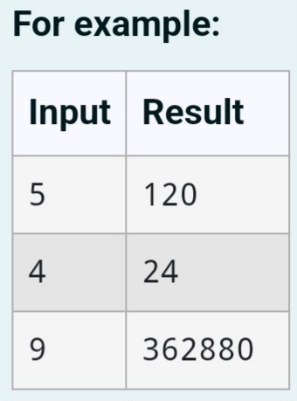
6. Write a program to find the factorial of a given number.

The given number will be passed to the program as an input of type int.

The program is expected to calculate the factorial of the given number and return it as an int type.

Assumptions for this program:

The given input number will always be greater than or equal to 1.

****Due to the range supported by int. The input numbers will range from 1 to 12.

**Program:**

n=int(input())

a=1

for i in range(1,n+1):

a \*= i

print(a)

**Output:**

7. Write a program to find the count of the number of prime numbers in a specified range.



**Program:**

a=int(input())

b=int(input())

c=0

for num in range(a, b + 1):

if num > 1:

for i in range(2, int(num \*\* 0.5) + 1):

if (num % i) == 0:

break

else:

c += 1

print(c)

**Output:**

8. Rakesh loves playing with numbers. He took the Fibonacci series and wants to find the sum of squares of the series until a given value. Write a code that implements his task.

Input Format: Single Integer N

Output Format: Display the sum of squares of the Fibonacci series until the Nth term.

Example

**Input**: 9

**Output**: 1870

Explanation:

The numbers are: 1 1 2 3 5 8 13 21 34

Sum of their squares is: 1 +1+4+9+25 + 64+169+441+1156 = 1870

**Program:**

n= int(input())

a, b = 0, 1

s = 0

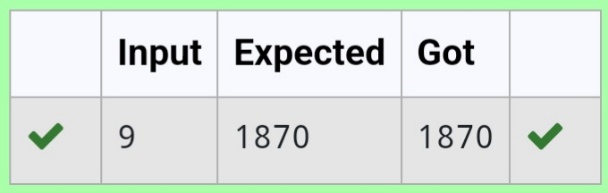
for \_ in range(n):

s += b \*\* 2

a, b = b, a + b

print(s)

**Output:**

****

9. Write a program to find the sum of the series 1+11+111+1111+...+ n terms (n will be given as input from the user and sum will be the output)

Sample Test Cases

Test Case 1

**Input**

4

**Output**

1234

**Program:**

n = int(input())

term = 1

sum\_series = 0

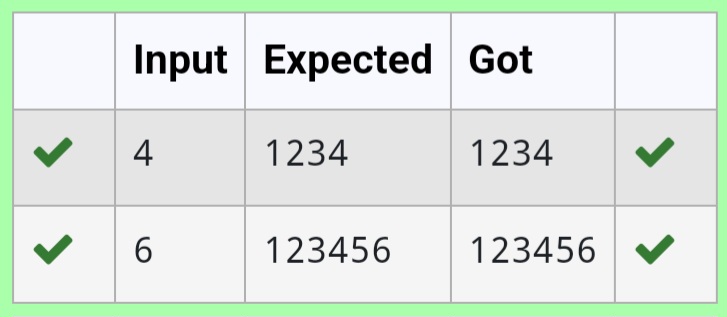
for i in range(n):

sum\_series += term

term = term \* 10 + 1

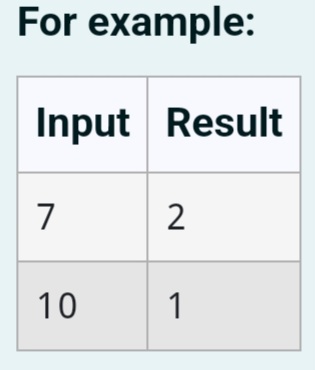
print(sum\_series)

**Output:**

****

10. Write a program that finds whether the given number N is Prime or not.

If the number is prime, the program should return 2 else it must return 1.



**Program:**

N = int(input())

if N < 2:

result = 1

else:

for i in range(2, int(N\*\*0.5) + 1):

if N % i == 0:

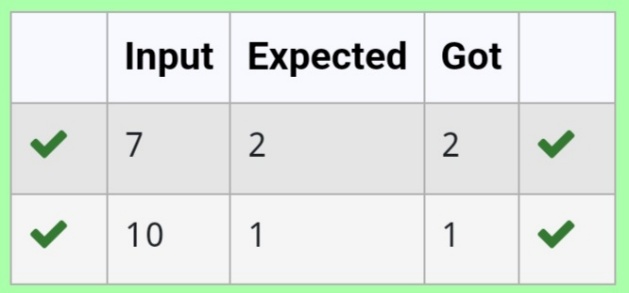
result = 1

break

else:

result = 2

print(result)

**Output:**