WEEK 4-

1. Given a number N, find the next perfect square greater than N.

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

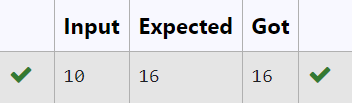
**PROGRAM:**

n=int(input())

ns=(int(n\*\*0.5)+1)\*\*2

print(ns)

**OUTPUT:**

****

2. 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 the sum will be the output)

INPUT:

4

OUTPUT:

1234

**PROGRAM:**

n=int(input())

term=1

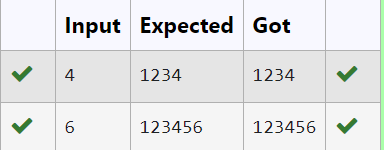
sos=0

for i in range(n):

sos+=term

term=term\*10+1

print(sos)

**OUTPUT:**

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

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

NOTE: Fibonacci series looks like -

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55,... and so on.

i.e. Fibonacci series starts with 0 and 1, and continues generating the next number as the sum of the previous two numbers.

• first Fibonacci number is 0,

• second Fibonacci number is 1,

• third Fibonacci number is 1,

• fourth Fibonacci number is 2,

• fifth Fibonacci number is 3,

• sixth Fibonacci number is 5,

• seventh Fibonacci number is 8, and so on.

**PROGRAM:**

N=int(input())

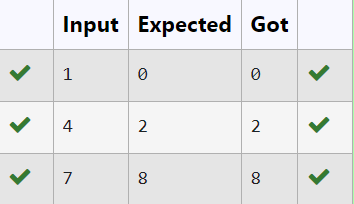
a,b=0,1

for x in range(n-1):

a,b=b,a+b

print(a)

**OUTPUT:**

****

4. 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.

**PROGRAM:**

n=input()

c=0

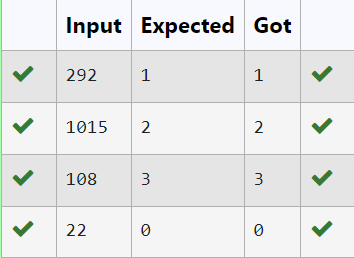
for digit in set(n):

if num.count(digit)==1:

c+=1

print(c)

**OUTPUT:**

****

5. Given a positive integer N,check whether it can be represented as a product of single digit numbers.

INPUT:

14

OUTPUT:

Yes

**PROGRAM:**

n=int(input())

if n<10:

print("Yes")

else:

for i in range(2,10):

while n%i==0:

n//=i

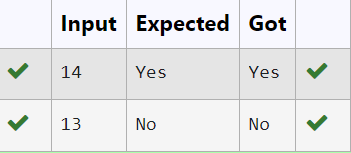
if n==1:

print("Yes")

else:

print("No")

**OUTPUT:**

****

6. 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

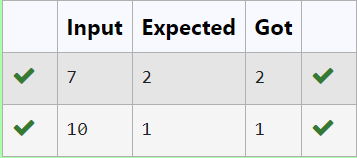
break

else:

result=2

print(result)

**OUTPUT:**

****

7. Determine the factors of a number

**PROGRAM :**

Num=int(input())

For x in range(1,num+1):

if num%x==0:

print(x,end=” ”)

**OUTPUT:**

8. Given an integer N,check whether N the given number can be made a perfect square after adding 1 to it .

**PROGRAM:**

Import math

Num=int(input())

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

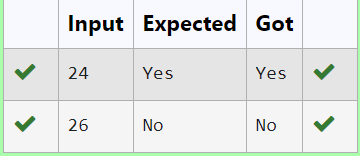
If num+1==Sqrt\_num\*Sqrt\_num:

Print(“Yes”)

Else :

Print(“No”)

**OUTPUT:**

****

9. 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'.

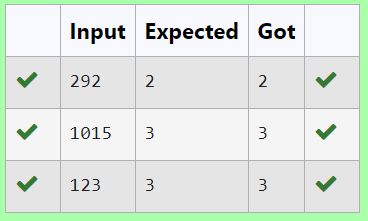
**PROGRAM:**

n=input()

c=len(set(n))

print(c)

**OUTPUT:**

****

10. A number is said to be disarium number when the sum of the digits raised to the power of thir respective positions becomes equal to the number itself . Write a program to print number is disarium or not.

**PROGRAM :**

n=input()

sod=sum(int(digit)\*\*(index+1) for index,digit in enumerate(n))

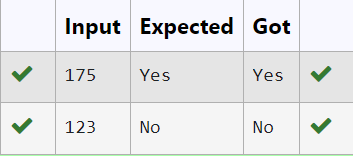
if sod==int(n):

print(“Yes”)

else:

print(“No”)

**OUTPUT:**

****