In mathematics, the factorial of a non-negative integer n, denoted by n!, is the product of all positive integers less than or equal to n. For example,

5! = 5 \* 4 \* 3 \* 2 \* 1 = 120

4! = 4 \* 3 \* 2 \* 1 = 24

9! = 9 \* 8 \* 7 \* 6 \* 5 \* 4 \* 3 \* 2 \* 1 = 362880

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

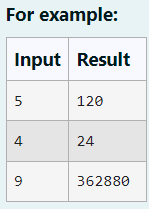
The given number will be passed to the program as an input of type int.

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:**

a=int(input())

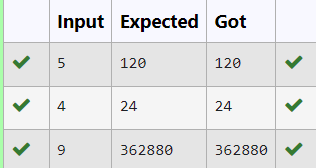
s=1

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

s=s\*i

print(s)

**Output:**

****

2. 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: 1123 5 8 13 21 34

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

For example:

Input

9

Result

1870

**Program:**

n=int(input())

a, b = 0, 1

s = 0

for i in range(n):

s = s + (b \*\*2)

a, b=b, a+b

print(s)

**Output:**

****

3. An abundant number is a number for which the sum of its proper divisors is greater than the number itself.

Proper divisors of the number are those that are strictly lesser than the number.

Input Format:

Take input an integer from stdin

Output Format:

Print Yes if given number is Abundant. Otherwise, print No

Example input:

12

Output:

Yes

Explanation

The proper divisors of 12 are: 1, 2, 3, 4, 6, whose sum is 1 + 2 +3+4+6 16. Since sum of proper divisors is greater than the given number, 12 is an abundant number.

Example input:

13

Output:

No

Explanation

The proper divisors of 13 is: 1, whose sum is 1. Since sum of proper divisors is not greater than the given number, 13 is not an abundant number.

**Program:**

n=int(input())

s=0

for i in range(1,n):

if (n%i==0):

s=s+i

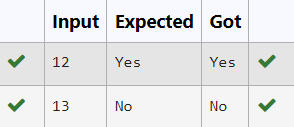
if (s>n):

print(“Yes”)

else:

print(“No”)

**Output:**

****

4. 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

Test Case 2

Input

6

Output

123456

**Program:**

N=int(input())

Term=1

Series=0

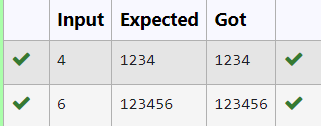
for i in range(N):

Series+=Term

Term=Term\*10+1

Print(Series)

**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

n=int(input())

x=n+1

z=math.sqrt(x)

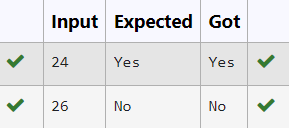
if (z==int(z)):

print(“Yes”)

else:

print(“No”)

**Output:**

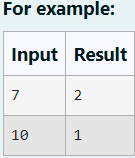
****

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.

Assumption: 2<=N<=5000, where N is the given number.

Example 1: if the given number N is 7, the method must return 2

Example 2: if the given number N is 10, the method must return 1



**Program:**

n=int(input())

a=0

if(2<=n<=5000):

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

if(n%i==0):

a+=1

else:

result=1

if(a<=2):

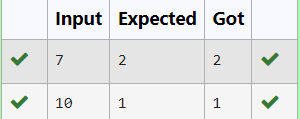
result=2

else:

result=1

print(result)

**Output:**

****

7. 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,

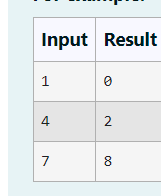
• third Fibonacci number is 1,

• 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())

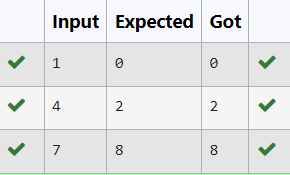
a,b=0,1

for i in range(n-1):

a,b=b,a+b

print(a)

**Output:**

****

**8.** 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 Input:

175

Output:

Yes

Explanation

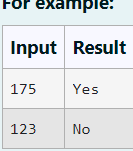
1^1+7^2+5^3 = 175

Example Input:

123

Output:

No



**Program:**

num=input()

n=len(num)

res=0

for i,digit in enumerate(num):

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

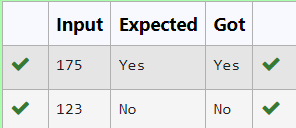
if res==int(num):

print(“Yes”)

else:

print(“No”)

**Output:**

****

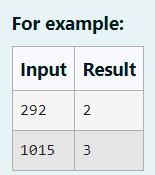
9. Write a program to find the count of unique digits in a given number N. The number will be passed to the program as an input of type int.

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'.



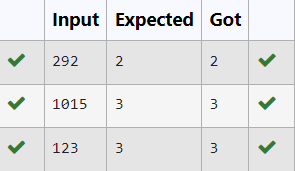
**Program:**

N=input()

C=len(set(N))

Print(C)

**Output:**

****

10. 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:**

import math

n=int(input())

z=n+1

while z>0:

m=math.sqrt(z)

if(m==int(m)):

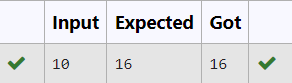
print(z)

break

else:

z=z+1

**Output:**

****