Ex. No. : 4.1 **Date: 12.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Factors of a number

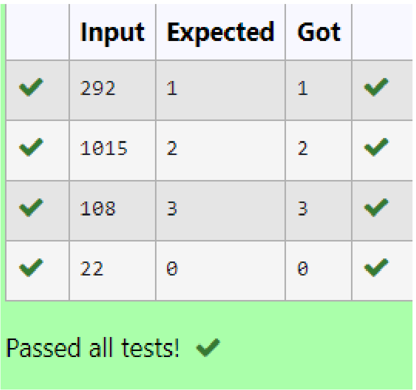
Determine the factors of a number (i.e., all positive integer values that evenly divide into a number).

For example:

Input	Result
292	1
1015	2
108	3
22	0

PROGRAM

```
def digits (n) :

count=0

for digit in range (10) :

if str(n).count(str(digit))==1:

count+=1

return count

n=int (input ())

print( digits (n) )
```



	Input	Expected	Got	
✓	292	1	1	✓
✓	1015	2	2	✓
✓	108	3	3	✓
✓	22	0	0	✓

Passed all tests! ✓

Ex. No. : 4.2 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Non Repeated Digit Count

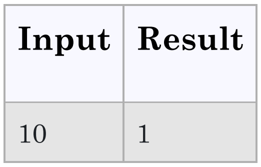
Write a program to find the count of non-repeated digits in a given number N. The number will be passed to the program as an input of type int.
Assumption: The input number will be a positive integer number >= 1 and <= 25000.
Some examples are as below.
If the given number is 292, the program should 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.

Example1: if the given number N is 7, the method must return 2

Example2: if the given number N is 10, the method must return 1

For example:

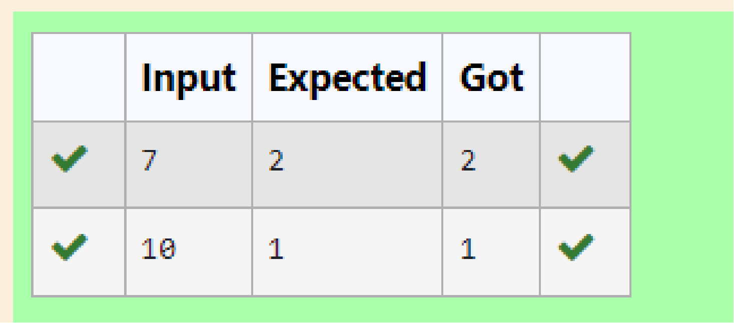
Input	Result
7	2



Input	Result
10	1

Program

```
n=int (input ( ))
f=0
 for i in range (2, n) :
if n%i==0:
f=1
Break
 if f==1:
print(1)
 else :
print(2)
```



	Input	Expected	Got	
✓	7	2	2	✓
✓	10	1	1	✓

Ex. No. : 4.3 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Prime Checking

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.

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

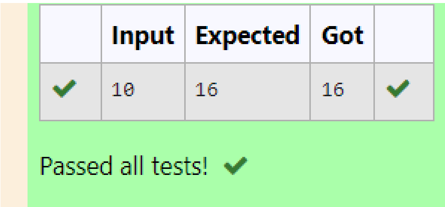
10

Output:

16

Program

```
def square (n) :
if(n%4==0):
return "Yes"
else:
return "No"
 n=int (input ( ))
print (square (n))
```



	Input	Expected	Got	
✓	10	16	16	✓

Passed all tests! ✓

Ex. No. : 4.4 **Date:14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Next Perfect Square

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

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.

For example:

Input:

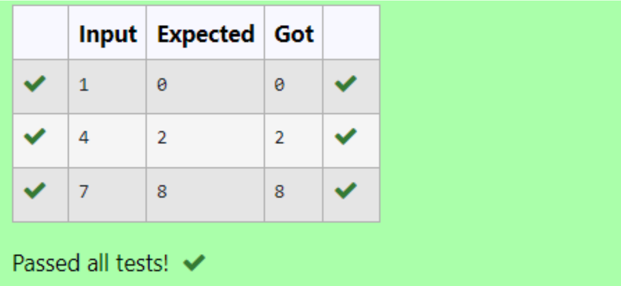
7

Output

8

Program

```
import math

n=int(input ())

n1=math.sqrt(n)

n2=n1+1

m=int (n2)

print (m**2)
```



	Input	Expected	Got	
✓	1	0	0	✓
✓	4	2	2	✓
✓	7	8	8	✓

Passed all tests! ✓

Ex. No. : 4.5 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Nth Fibonacci

Write a program to return the nth number in the fibonacci series. The value of N will be passed to the program as input.

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:

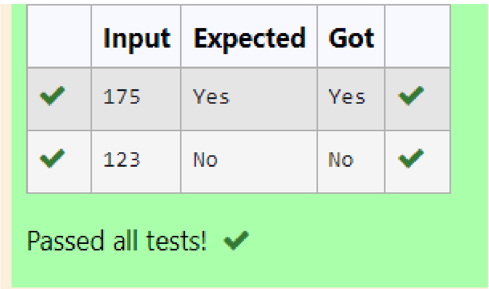
No

For example:

InputResult	
175	Yes
123	No

Program

```
n=int (input ())
if n<2:
print (n-1)
else:
n=n-1
 fs= [0,1]
for i in range (1, n) :
print(fs [n])
s.append(fs[i]+fs[i-1])
print(fs[n])
```



	Input	Expected	Got	
✓	175	Yes	Yes	✓
✓	123	No	No	✓

Passed all tests! ✓

Ex. No. : 4.6 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Disarium Number

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.

Sample Test Cases

Test Case 1

Input

4

Output

1234

Explanation:

as input is 4, have to take 4 terms.

1 + 11 + 111 + 1111

Test Case 2

Input

6

Output

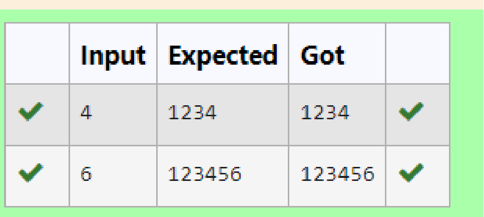
123456

For example:

Input	Result
3	123

Program

```
import math
 a=int (input ( ))
b= (pow(10, a+1) -10-9*a) /81
print(' {:.0f}'.format (b))
```



	Input	Expected	Got	
✓	4	1234	1234	✓
✓	6	123456	123456	✓

Ex. No. : 4.7 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Sum of Series

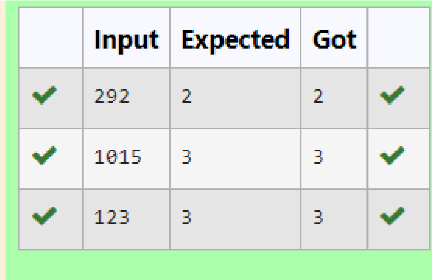
Write a program to find the sum of the series 1 +11 + 111 + 1111 + . . . + n terms (n will be given as input from the user and sum will be the output)

For example:

Input	Result
292	2
1015	3

Program

```
Def digit(N):
Return len(set(str(N)))
N=int(input())
Print(digit(N))
```



	Input	Expected	Got	
✓	292	2	2	✓
✓	1015	3	3	✓
✓	123	3	3	✓

Ex. No. : 4.8 **Date:14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Unique Digit Count

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'.

Input Format:

Single Integer input.

Output Format:

Output displays Yes if condition satisfies else prints No.

Example Input:

14

Output:

Yes

Example Input:

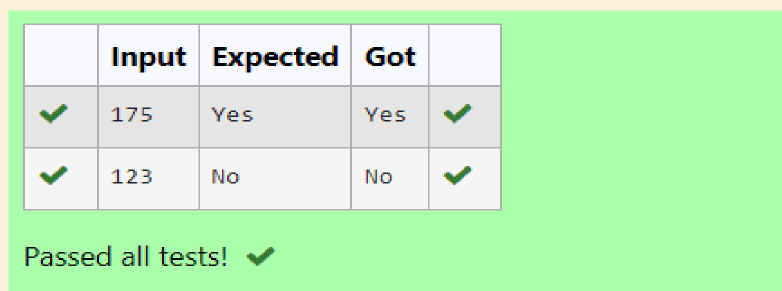
13

Output:

No

Program

```
a=int (input ())

if a%2==0 or a%3=-0 or a%5==0 or a%7==0 or a%9==0:

print("Yes")

 else:

print ("No")
```



	Input	Expected	Got	
✓	175	Yes	Yes	✓
✓	123	No	No	✓

Passed all tests! ✓

Ex. No. : 4.9 **Date: 14.04.24**

Register No.: 231401029 **Name: Ernita Poobalarayan**

Product of single digit

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

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

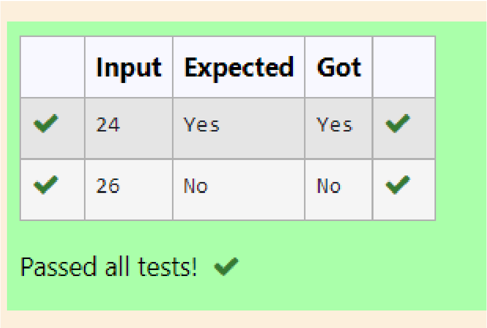
No

For example:

Input	**Result**
24	Yes

Program

```
def square(n) :

if (n%4=-0):

return

else

return ‘No’

n=int (input ())

print (square (n))
```



	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	No	✓

Passed all tests! ✓

Ex. No. : 4.10 **Date:14.04.24**

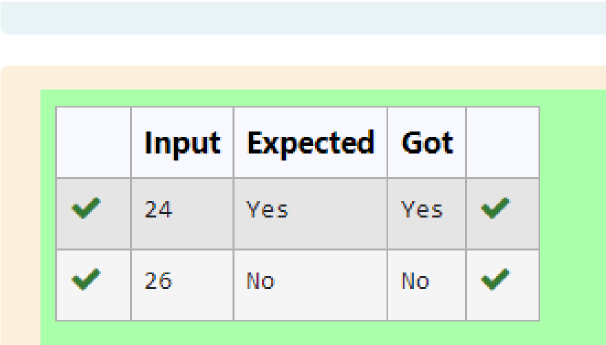
Register No.: 231401029 **Name: Ernita Poobalarayan**

Perfect Square After adding One

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

Program

```
def square (n) :
if(n%4==0):
return "Yes"
else:
return "No"
 n=int (input ( ))
print (square (n))
```



	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	No	✓