

04 - Iteration Control Structures

For example:

Input	Result
20	1 2 4 5 10 20

Ex. No. : 4.1

Date:

Register No.:

Name:

Factors of a number

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

```
a=int(input())
for i in range(1,a+1):
    if(a%i==0):
        print(i,end=" ")
```

For example:

Input	Result
292	1
1015	2
108	3
22	0

Ex. No. : 4.2

Date:

Register No.:

Name:

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.

```
a=(input())
b={}
for i in a:
    if(i in b):
        b[i]+=1
    else:
        b[i]=1
c=0
for i in b.values():
    if(i==1):
        c=c+1
print(c)
```

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
10	1

Ex. No. : 4.3

Date:

Register No.:

Name:

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: $2 \leq N \leq 5000$, where N is the given number.

```
a=int(input())
b=0
for i in range(1,a):
    if(a%i==0):
        b+=1
if(b>2):
    print("1")
else:
    print("2")
```

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

Ex. No. : 4.4

Date:

Register No.:

Name:

Next Perfect Square

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

```
h=int(input())
d=0
for i in range(1,h):
    if(i*i>h):
        d=i*i
        break
print(d)
```

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

Ex. No. : 4.5

Date:

Register No.:

Name:

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.

```
a=int(input())
b=1
temp=0
c=0
for i in range(0,a):
    c=temp+b
    b=temp
    temp=c
print(b)
```

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

Ex. No. : 4.6

Date:

Register No.:

Name:

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.

```
n=int(input())
num=len(str(n))
l=num
a=n
d=0
sum=0
for i in range(0,num):
    d=n%10
    sum=sum+d**l
    l=l-1
    n=n//10
if sum==a:
    print("Yes")
else:
    print("No")
```

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

Ex. No. : 4.7

Date:

Register No.:

Name:

Sum of Series

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

```
a=int(input())
sum=1
count=1
for i in range(1,a):
    sum=sum*10+1
    count=count +sum
print(count)
```

For example:

Input	Result
292	2
1015	3

Ex. No. : 4.8

Date:

Register No.:

Name:

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'.

```
a=int(input())
b=[]
while(a>0):
    if(a%10 not in b):
        b.append(a%10)
    a=a//10
print(len(b))
```

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

Ex. No. : 4.9

Date:

Register No.:

Name:

Product of single digit

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

```
a=int(input())
if(a%2==0 or a%3==0 or a%5==0 or a%7==0):
    print("Yes")
else:
    print("No")
```

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

Ex. No. : 4.10

Date:

Register No.:

Name:

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.

```
a=int(input())
a=a+1
b=0
if(a==0 or b==1):
    b=1
for i in range(2,(a//2)):
    if(a==i*i):
        b=1
        break
if(b==1):
    print("Yes")
else:
    print("No")
```