

## **04 - Iteration Control Structures**

Ex. No. : 4.1

Date: 17.04.2024

Register No.: 231501165

Name: SURWEESH SP

## 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	
20	1 2 4 5 10 20	

```
k=int(input())
```

```
l=[]
```

```
for i in range(1,k+1):
```

```
    if(k%i==0):
```

```
        l.append(i)
```

```
for j in l:
```

```
    print(j,end=' ')
```

	Input	Expected	Got	
✓	20	1 2 4 5 10 20	1 2 4 5 10 20	✓
✓	5	1 5	1 5	✓
✓	13	1 13	1 13	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 4.2

Date: 17.04.2024

Register No.: 231501165

Name: SURWEESH SP

## 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  $\geq 1$  and  $\leq 25000$ . 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.

**For example:**

Input	Result
292	1
1015	2
108	3
22	0

```
n=int(input())
l=[]
k=[]
while n>0:
    a=n%10
    n=n//10
    l.append(a)
for i in range(len(l)):
    if l.count(l[i])==1:
```

```
k.append(l[i])  
print(len(k))
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 4.3**

**Date: 17.04.2024**

**Register No.: 231501165**

**Name: SURWEESH SP**

## **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.

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

```
a=int(input())
for i in range(2,a):
    if(a%i==0):
        flag=0
    elif(a%i!=0):
        flag=1
    else:
        flag=0
if(flag==1):
    print("2")
elif(flag==0):
    print("1")
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 4.4

Date: 24.04.2024

Register No.: 231501165

Name: SURWEESH SP

## Next Perfect Square

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

```
a=int(input())
```

```
c=[]
```

```
for i in range(0,a):
```

```
    b=i**2
```

```
    if(b>a):
```

```
        c.append(b)
```

```
print(c[0])
```

	Input	Expected	Got	
✓	10	16	16	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 4.5**

**Date: 24.04.2024**

**Register No.: 231501165**

**Name: SURWEESH SP**

## **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.

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

```
a=[0,1]
```

```
for i in range(0,100):
```

```
    a.append(a[-1]+a[-2])
```

```
q=int(input())
```

```
print(a[q-1])
```



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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 4.6

Date: 24.04.2024

Register No.: 231501165

Name :SURWEESH SP

## 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.

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:**

Inp ut	Res ult
175	Yes
123	No

```
import math
```

```
n=int(input())
a=len(str(n))
sum=0
x=n
while(x!=0):
    r=x%10
    sum=int(sum+math.pow(r,a))
    a-=1
    x=x//10
if(sum==n):
    print("Yes")
else:
    print("No")
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 4.7**

**Date: 24.04.2024**

**Register No.: 231501165**

**Name: SURWEESH SP**

## **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)

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

```
n=int(input())
```

```
b=1
```

```
sum=0
```

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

```
    sum+=b
```

```
    b=(b*10)+1
```

```
print(sum)
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

Ex. No. : 4.8

Date: 24.04.2024

Register No.: 231501165

Name: SURWEESH SP

## 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  $\geq 1$  and  $\leq 25000$ .

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'.

**For example:**

Input	Result
292	2
1015	3

```
a=int(input())
```

```
b=[]
```

```
while a>0:
```

```
    c=a%10
```

```
    a=a//10
```

```
    b.append(c)
```

```
b=list(set(b))
```

```
print(len(b))
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 4.9**

**Date: 24.04.2024**

**Register No.: 231501165**

**Name: SURWEESH SP**

## **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:

Output displays Yes if condition satisfies else prints No.

Example Input:

14

Output:

Yes

Example Input:

13

Output:

No

```
a=int(input())
```

```
flag=0
```

```
for i in range(10):
```

```
    for j in range(10):
```

```
        if(i*j==a):
```

```
            flag=1
```

```
            break
```

```
if(flag==1):
```

```
    print("Yes")
```

```
else:
```

```
    print("No")
```



	Input	Expected	Got	
✓	14	Yes	Yes	✓
✓	13	No	No	✓

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.

**Ex. No. : 4.10**

**Date: 1.5.2024**

**Register No.: 231501165**

**Name: SURWEESH SP**

## **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.

Input Format:

Single integer input.

Output Format:

Yes or No.

Example Input:

24

Output:

Yes

Example Input:

26

Output:

No

**For example:**

<b>Input</b>	<b>Result</b>
24	Yes

```
import math
```

```
n=int(input())
```

```
a=n+1
```

```
sr=int(math.sqrt(a))
```

```
if(sr*sr==a):  
    print("Yes")  
else:  
    print("No")
```

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

Passed all tests! ✓

**Correct**

Marks for this submission: 1.00/1.00.