

04 - Iteration Control Structures

Ex. No. : 4.1

Date: 13.04.

Register No.: 231901020

Name: KAVIYA.V

Factors of a number

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

CODING:

```
n=int(input())
f=[]
for i in range(1,n+1):
    if n%i==0:
        f.append(i)
print(*f)
```

For example:

Input	Result
20	1 2 4 5 10 20

	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	✓

Ex. No. : 4.2

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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.

CODING:

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

```
nstr=str(n)
```

```
dc={}
```

```
for d in nstr:
```

```
    if d in dc:
```

```
        dc[d]+=1
```

```
    else:
```

```
        dc[d]=1
```

```
nsc=0
```

```
for d,c in dc.items():
```

```
    if c==1:
```

```
        nsc+=1
```

```
print(nsc)
```

For example:

Input	Result
292	1
1015	2
108	3
22	0

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

Ex. No. : 4.3

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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.

CODING:

```
n=int(input())
if n<2:
    print(1)
    exit()
for i in range(2,int(n**0.5)+1):
    if n%i==0:
        print(1)
        exit()
print(2)
```

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

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. : 4.4

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

Next Perfect Square

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

CODING:

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

```
sn=int(n**0.5)
```

```
nps=(sn+1)**2
```

```
print(nps)
```

Input Format:

Integer input from stdin.

Output Format:

Perfect square greater than N.

Example Input:

10

Output:

16

	Input	Expected	Got	
✓	10	16	16	✓

Ex. No. : 4.5

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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.

CODING:

```
n=int(input())
f1=0
f2=1
if n==1:
    print(f1)
elif n==2:
    print(f2)
else:
    for i in range(2,n):
        fn=f1+f2
        f1=f2
        f2=fn
    print(fn)
```

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

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. : 4.6

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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.

CODING:

```
n=input()
sd=0
p=1
for d in n:
    sd+=int(d)**p
    p+=1
if sd==int(n):
    print("Yes")
else:
    print("No")
```

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

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. : 4.7

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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)

CODING:

```
n=int(input())
s=0
c=1
for i in range(n):
    s+=c
    c=c*10+1
print(s)
```

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

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. : 4.8

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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'.

CODING:

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

```
ud=set()
```

```
ns=str(n)
```

```
for i in ns:
```

```
    ud.add(i)
```

```
print(len(ud))
```

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

For example:

Input	Result
292	2
1015	3

Ex. No. : 4.9

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

Product of single digit

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

CODING:

```
n=int(input())
c=False
for i in range(1,10):
    if n%i==0 and n//i<10:
        c=True
        break
if c:
    print("Yes")
else:
    print("No")
```

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

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. : 4.10

Date: 13.04.24

Register No.: 231901020

Name: KAVIYA.V

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.

CODING:

```
n=int(input())
n+=1
sn=int(n**0.5)
if sn**2==n:
    print("Yes")
else:
    print("No")
```

	Input	Expected	Got	
✓	24	Yes	Yes	✓
✓	26	No	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