04 – Iteration Control Structures

Ex. No.: 4.1 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

Abundant Numbers

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:

Answer: (penalty regime: 0 %)

```
n=int(input())
s=0
for i in range(1,(n//2+1)):
    if(n%i==0):
        s=s+i
if(s>n):
    print("Yes")
else:
    print("No")
```

Output:

	Input	Expected	Got	
✔	12	Yes	Yes	✔
✔	13	No	No	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.2 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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'.

For example:

Input	Result
292	2
1015	3

Program:

```
n=int(input())
l=list(set(str(n)))
print(len(l))
```

Output:

	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.: 4.3 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: 2 <=starting number of the range<= ending number of the range<=7919

Example1: If the starting and ending number or the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2. 3. 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

For example:

Input	Result
2 20	8
700 725	3

Program:

```
st=int(input())
en=int(input())
c=0
while(st<=en):
    f=0
    for i in range(2,st):
        if(st%i==0):
            f=1
    if(f==0):
        c=c+1
    st=st+1
print(c)
```

Output:

	Input	Expected	Got	
✔	2 20	8	8	✔
✔	700 725	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.4 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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:

```
n=int(input())
sq=int(n**0.5)
print((sq+1)**2)
```

Output:

	Input	Expected	Got	
✔	10	16	16	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.5 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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.

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

Program:

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

Output:

	Input	Expected	Got	
✔	7	2	2	✔
✔	10	1	1	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.6 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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: 1 1 2 3 5 8 13 21 34

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

For example:

Input	Result
9	1870

Program:

```
n=int(input())
a=b=1
s=2
for i in range(2,n):
    c=a+b
    s=s+c**2
    a=b
    b=c
print(s)
```

Output:

	Input	Expected	Got	
✔	9	1870	1870	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.7 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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.

For example:

Input	Result
292	1
1015	2
108	3
22	0

Program:

```
n=input()
count=0
dic={}
for i in  n:
    dic[i]=dic.get(i,0)+1
for j in dic.values():
    if j==1:
        count+=1
print(count)

```

Output:

	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.: 4.8 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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:

Input	Result
175	Yes
123	No

Program:

```
n=int(input())
l=list(str(n))
s=0
for i in range(len(l)):
    a=int(l[i])
    s=s+a**(i+1)
if(s==n):
    print("Yes")
else:
    print("No")
```

Output:

	Input	Expected	Got	
✔	175	Yes	Yes	✔
✔	123	No	No	✔

Passed all tests! ✔

Ex. No.: 4.9 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

Write a program to find the count of the number of prime numbers in a specified range.

The starting and ending number of the range will be provided as input to the program.

Assumption: 2 <=starting number of the range<= ending number of the range<=7919

Example1: If the starting and ending number or the range is given as 2 and 20, the program must return 8, because there are 8 prime numbers in the specified range from 2 to 20. namely (2. 3. 5, 7, 11, 13, 17, 19)

Example2: If the starting and ending number of the range is given as 700 and 725, the program must return 3, because there are 3 prime numbers in the specified range from 700 to 725, namely (701, 709, 719)

For example:

Input	Result
2 20	8
700 725	3

Program:

```
st=int(input())
en=int(input())
c=0
while(st<=en):
    f=0
    for i in range(2,st):
        if(st%i==0):
            f=1
    if(f==0):
        c=c+1
    st=st+1
print(c)
```

Output:

	Input	Expected	Got	
✔	2 20	8	8	✔
✔	700 725	3	3	✔

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.

Ex. No.: 4.10 Date:

Register No.: 2116231501105 Name: Nandhini Prakash

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	Result
1	0
4	2

Program:

```
n=int(input())
a=0
b=1
if(n==1):
    print(a)
elif(n==2):
    print(b)
else:
    s=0
    s=a+b
    for i in range(2,n):
        c=a+b
        a=b
        b=c
    print(c)

```

Output:

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

Passed all tests! ✔

Correct

Marks for this submission: 1.00/1.00.