

1. Python program to count number of trailing zeros in Factorial of number N

Sample input/output:

Input: n = 5

Output: 1

Factorial of 5 is 120 which has one trailing 0.

Input: n = 20

Output: 4

Factorial of 20 is 2432902008176640000 which has
4 trailing zeroes.

Input: n = 100

Output: 24

```
In [2]: n=int(input("The number to be factorized:"))
x = n // 5
y = x
while x > 0:
    x /= 5
    y += int(x)

print('Number of trailing zeroes:', y)
```

```
The number to be factorized:20
Number of trailing zeroes: 4
```

2. Write a Python Program to Find largest prime factor of a number.

Given a positive integer (1 $\leq n \leq 10^{15}$). Find the largest prime factor of a number.

Input: 6

Output: 3

Explanation

Prime factor of 6 are- 2, 3

Largest of them is '3'

Input: 15

Output: 5

```

In [4]: import math
n=int(input())

    # Initialize the maximum prime factor
    # variable with the lowest one
maxPrime = -1

    # Print the number of 2s that divide n
while n % 2 == 0:
    maxPrime = 2
    n >>= 1    # equivalent to n /= 2

    # n must be odd at this point
while n % 3 == 0:
    maxPrime = 3
    n=n/3

    # now we have to iterate only for integers
    # who does not have prime factor 2 and 3
for i in range(5, int(math.sqrt(n)) + 1, 6):
    while n % i == 0:
        maxPrime = i
        n = n / i
    while n % (i+2) == 0:
        maxPrime = i+2
        n = n / (i+2)

    # This condition is to handle the
    # case when n is a prime number
    # greater than 4
if n > 4:
    maxPrime = n
print(int(maxPrime))

```

6
3

3. Given three angles, a, b and c, write a Python program to check if it is possible to have a triangle of positive area with these angles. If it is possible print “YES” else print “NO”.

The 3 angles form a triangle with positive area if a, b and c are not 0 and the sum of a, b and c is 180 and $(a + b) \geq c$ or $(b + c) \geq a$ or $(a + c) \geq b$.

Examples:

Input : ang1 = 50, ang2 = 60, ang3 = 70

Output : YES

Input : ang1 = 50, ang2 = 65, ang3 = 80

Output : NO

```
In [7]: a = int(input('Enter a: '))
b = int(input('Enter b: '))
c = int(input('Enter c: '))

if a != 0 and b != 0 and c != 0 and a + b + c == 180:
    if a+b >= c or b+c >= a or a+c >= b:
        print('YES')
    else:
        print('NO')
else:
    print('NO')
```

```
Enter a: 50
Enter b: 65
Enter c: 80
NO
```

4. Python program to print all happy numbers between 1 and 100

A happy number is a number which eventually reaches 1 when replaced by the sum of the square of each digit.

1) Take any two-digit number: **13**

2) Square the digits, then add:

$$1^2 + 3^2 = 1 + 9 = 10$$

3) Repeat step 2:

$$1^2 + 0^2 = 1$$

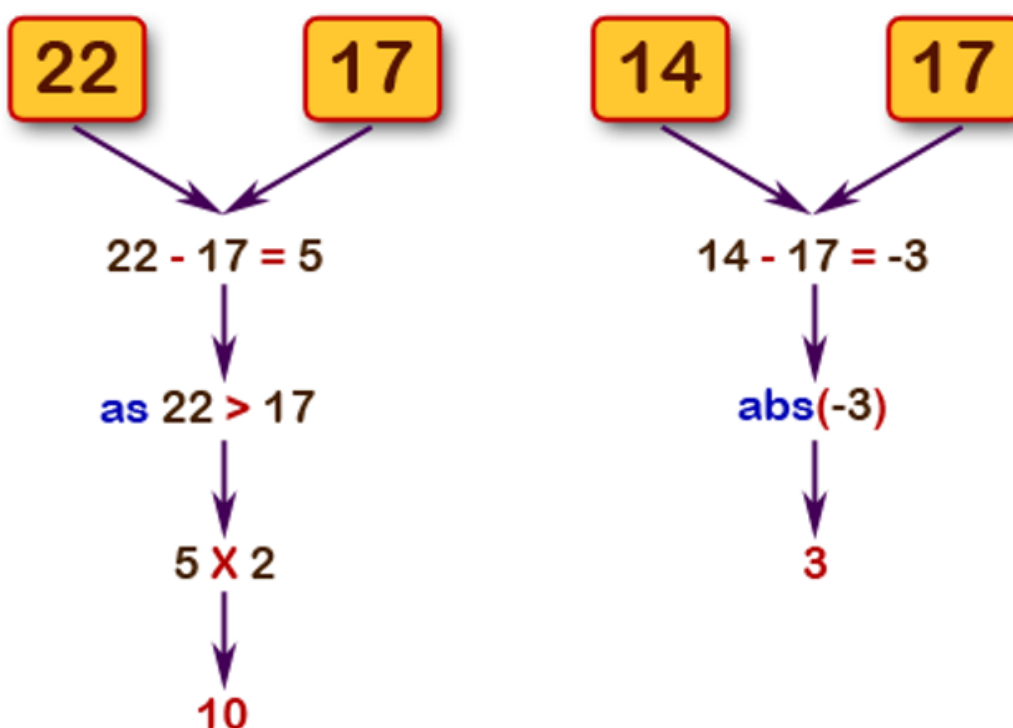
13 is a happy number!
A number is happy if
it eventually reaches 1.



```
In [ ]: #n = int(input('Enter n: '))
for n in range(1, 101):
    temp = n
    while n!= 1 and n!= 4:
        sum=0
        s = str(n)
        for i in s:
            sum += int(i)**2
        n = sum
    if n==1:
        print(temp)
```

1
7
10
13
19
23
28
31
32
44
49
68
70
79
82
86
91
94
97
100

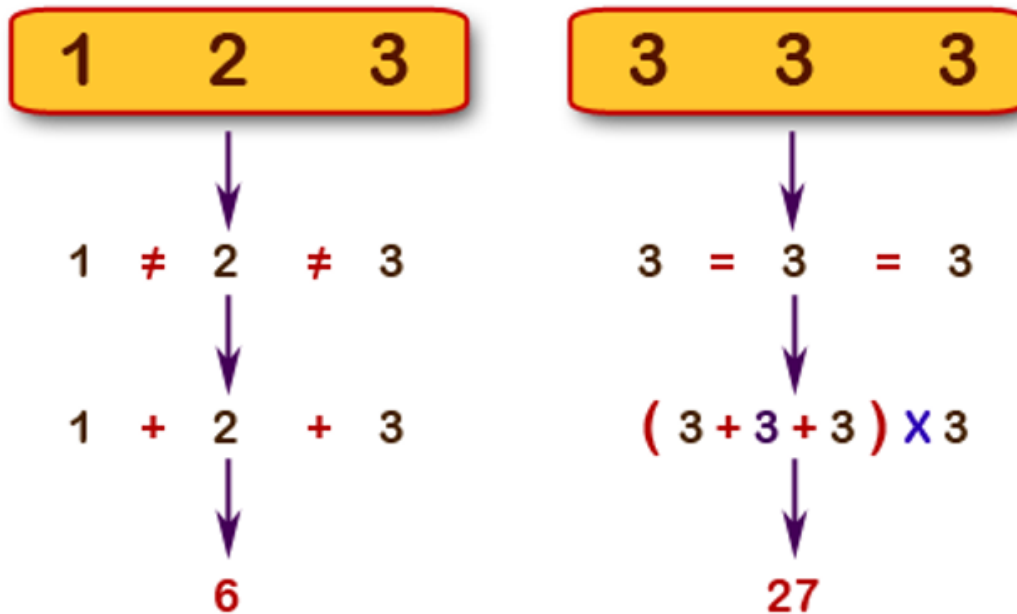
5. Write a Python Program to Get the difference between a given number and 17, if the number is greater than 17 return double the absolute difference



```
In [9]: n=int(input("Enter the number whose difference is to be found:"))
if n <= 17:
    print(17-n)
else:
    print((n - 17) * 2)
```

Enter the number whose difference is to be found:14
3

6. Write a Python program to calculate the sum of three given numbers, if the values are equal then return thrice their sum.



```
In [11]: x=int(input("Enter x:"))
y=int(input("Enter y:"))
z=int(input("Enter z:"))
sum = x + y + z
if x == y == z:
    sum = sum * 3
print(sum)
```

Enter x:3
Enter y:3
Enter z:3
27

7. Write a Python program to compute the greatest common divisor (GCD) of two positive integers.

The greatest common divisor (GCD) of two nonzero integers a and b is the greatest positive integer d such that d is a divisor of both a and b ; that is, there are integers e and f such that $a = de$ and $b = df$, and d is the largest such integer. The GCD of a and b is generally denoted $\text{gcd}(a, b)$. For example, the greatest common factor of 15 and 10 is 5, since both the numbers can be divided by 5.

```
In [12]: x=int(input("Enter the 1st number :"))
y=int(input("Enter the 2nd number:"))
gcd = 1
if x % y == 0:
    print(y)
for k in range(int(y / 2), 0, -1):
    if x % k == 0 and y % k == 0:
        gcd = k
        break
print(gcd)
```

```
Enter the 1st number :15
Enter the 2nd number:10
5
```

8. Write a Python program to find the median among three given numbers.




```
In [13]: x = input("Input the first number")
y = input("Input the second number")
z = input("Input the third number")
print("Median of the above three numbers -")

if y < x < z:
    print(x)
elif z < x < y:
    print(x)

elif z < y < x:
    print(y)
elif x < y < z:
    print(y)

elif y < z < x:
    print(z)
elif x < z < y:
    print(z)
```

```
Input the first number25
Input the second number15
Input the third number35
Median of the above three numbers -
25
```

9. Write a Python program to compute the amount of the debt in n months. The borrowing amount is \$100,000 and the loan adds 5% interest of the debt and rounds it to the nearest 1,000 above month by month.

Sample Input/Output:

An integer n ($0 \leq n \leq 100$)

Input number of months: 7

Amount of debt: \$144000

```
In [20]: n = int(input('Input number of months: '))

debt = 100000

while n > 0:
    debt *= 1.05
    if debt % 1000:
        debt = int(1+debt//1000)*1000
    n = n - 1

print('Amount of debt', debt)
```

Input number of months: 7
Amount of debt 144000

10. Write a Python program to compute and print sum of two given integers (more than or equal to zero). If given integers or the sum have more than 80 digits, print "overflow".

```
In [22]: x = int(input("Input first integer:"))
y = int(input("Input second integer:"))
if x >= 10 ** 80 or y >= 10 ** 80 or x + y >= 10 ** 80:
    print("Overflow!")
else:
    print("Sum of the two integers: ",x + y)
```

Input first integer:123
Input second integer:456
Sum of the two integers: 579

11. Write a Python program to compute the product of the odd digits in a given number, or 0 if there aren't any.

Input:
123456789
Output:
945

Input:
2468
Output:
0

Input:
13579
Output:
945

```
In [24]: n=int(input())
prod=1
if(int(c)%2 for c in str(n)):
    prod=1
    flag = 0
    for c in str(n):
        if int(c) % 2 == 1:
            flag = 1
            prod*= int(c)
    if flag == 0:
        print(0)
    else:
        print(prod)
```

123456

15

12. Write a Python program which iterates over the integers from 1 to 50. For multiples of three print "Fizz" instead of the number and for the multiples of five print "Buzz". For numbers which are multiples of both three and five print "FizzBuzz".

```
In [25]: for fizzbuzz in range(1,51):  
        if fizzbuzz % 3 == 0 and fizzbuzz % 5 == 0:  
            print("FizzBuzz")  
            continue  
        elif fizzbuzz % 3 == 0:  
            print("Fizz")  
            continue  
        elif fizzbuzz % 5 == 0:  
            print("Buzz")  
            continue  
        print(fizzbuzz)
```

```
1  
2  
Fizz  
4  
Buzz  
Fizz  
7  
8  
Fizz  
Buzz  
11  
Fizz  
13  
14  
FizzBuzz  
16  
17  
Fizz  
19  
Buzz  
Fizz  
22  
23  
Fizz  
Buzz  
26  
Fizz  
28  
29  
FizzBuzz  
31  
32  
Fizz  
34  
Buzz  
Fizz  
37  
38  
Fizz  
Buzz  
41  
Fizz  
43  
44  
FizzBuzz  
46  
47  
Fizz  
49  
Buzz
```

13. Write a Python program to calculate a dog's age in dog's years.

Note: For the first two years, a dog year is equal to 10.5 human years. After that, each dog year equals 4 human years.

Sample Input/Output:

```
Input a dog's age in human years: 12
The dog's age in dog's years is 61
```

```
In [26]: h_age = int(input("Input a dog's age in human years: "))

if h_age < 0:
    print("Age must be positive number.")
elif h_age <= 2:
    d_age = h_age * 10.5
else:
    d_age = 21 + (h_age - 2)*4

print("The dog's age in dog's years is", d_age)
```

```
Input a dog's age in human years: 12
The dog's age in dog's years is 61
```

14. Write a Python program to convert month name to number of days.

Sample Input/Output:

Input: January

Output: 31

Input: February

Output: 28

```
In [31]: print("List of months: January, February, March, April, May, June, July, August, September, October, November, December")
month_name = input("Input the name of Month: ")

if month_name == "February":
    print("No. of days: 28 days")
elif month_name == "April" or month_name == "June" or month_name == "September" or month_name == "November":
    print("No. of days: 30 days")
elif month_name == "January" or month_name == "March" or month_name == "May" or month_name == "July" or month_name == "August" or month_name == "October" or month_name == "December":
    print("No. of days: 31 days")
else:
    print("Wrong month name")
```

List of months: January, February, March, April, May, June, July, August, September, October, November, December
Input the name of Month: December
No. of days: 31 days

15. Write a Python program that reads an integer representing a month and prints the season for that month.

Pictorial Presentation:



```
In [32]: month = input("Input the month (e.g. January, February etc.): ")

if month == 'January' or month == 'February' or month == 'December':
    season = 'winter'
elif month == 'April' or month == 'May' or month == 'March':
    season = 'spring'
elif month == 'July' or month == 'August' or month == 'June':
    season = 'summer'
else:
    season = 'autumn'

print(season)
```

Input the month (e.g. January, February etc.): September
autumn

16. Write a Python program to get next day of a given date.

Sample:

Input a year: 1974

Input a month [1-12]: 2

Input a day [1-31]: 15

The next date is [yyyy-mm-dd] 1974-2-16.

```
In [33]: year = int(input("Input a year: "))

if (year % 400 == 0):
    leap_year = True
elif (year % 100 == 0):
    leap_year = False
elif (year % 4 == 0):
    leap_year = True
else:
    leap_year = False

month = int(input("Input a month [1-12]: "))

if month in (1, 3, 5, 7, 8, 10, 12):
    month_length = 31
elif month == 2:
    if leap_year:
        month_length = 29
    else:
        month_length = 28
else:
    month_length = 30

day = int(input("Input a day [1-31]: "))

if day < month_length:
    day += 1
else:
    day = 1
    if month == 12:
        month = 1
        year += 1
    else:
        month += 1
print('The next date is [yyyy-mm-dd]', year, '-', month, '-', day)
```

Input a year: 2024

Input a month [1-12]: 2

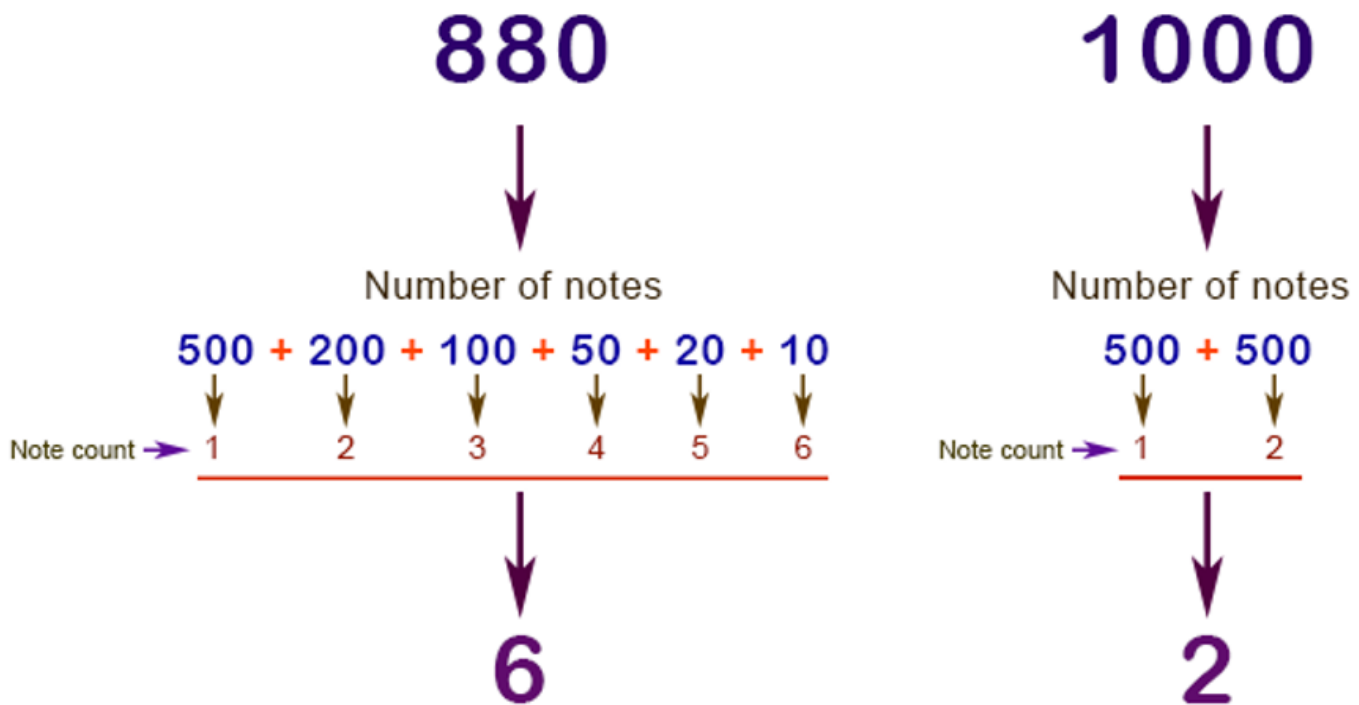
Input a day [1-31]: 28

The next date is [yyyy-mm-dd] 2024 - 2 - 29

17. Write a Python program to find the number of notes (Sample of notes: 10, 20, 50, 100, 200 and 500) against a given amount.

Range - Number of notes(n): n ($1 \leq n \leq 1000000$).

Pictorial Presentation:



```
In [ ]: money = int(input('Enter money: '))
notes = 0
notes += money//500
notes += (money%500)//200
notes += ((money%500)%200)//100
notes += (((money%500)%200)%100)//50
notes += ((((money%500)%200)%100)%50)//20
notes += ((((((money%500)%200)%100)%50)%20)//10
print(notes)
```

```
Enter money: 1000
2
```


18. Write a Python program to create a sequence where the first four members of the sequence are equal to one, and each successive term of the sequence is equal to the sum of the four previous ones. Find the Nth member of the sequence.

Example of the sequence: 1, 1, 1, 1, 4, 7, 13, ...

Sample input/output:

Input: 5

Output: 4

Input: 7

Output: 13

```
In [ ]: n=int(input("Enter n:"))
a=b=c=d=1
for i in range(1, n+1):
    if i <= 4:
        ans = 1
        #print(ans, end = ' ')
    else:
        ans = a+b+c+d
        #print(ans, end=' ')
        a = b
        b = c
        c = d
        d = ans
print(ans)
```

```
Enter n:11
181
```

19. Write a Python program to create maximum number of regions obtained by drawing n given straight lines.

If you draw a straight line on a plane, the plane is divided into two regions. For example, if you pull two straight lines in parallel, you get three areas, and if you draw vertically one to the other you get 4 areas.

Formula: number of regions = $((N^2 + N + 2) // 2)$

Where N is the number lines on a plane

Sample Input/Output:

Input number of straight lines:

5

Number of regions:

16

```
In [34]: while True:
          print("Input number of straight lines (o to exit): ")
          n=int(input())
          if n<=0:
              break
          print("Number of regions:")
          print((n*n+n+2)//2)
```

Input number of straight lines (o to exit):

1

Number of regions:

2

Input number of straight lines (o to exit):

2

Number of regions:

4

Input number of straight lines (o to exit):

5

Number of regions:

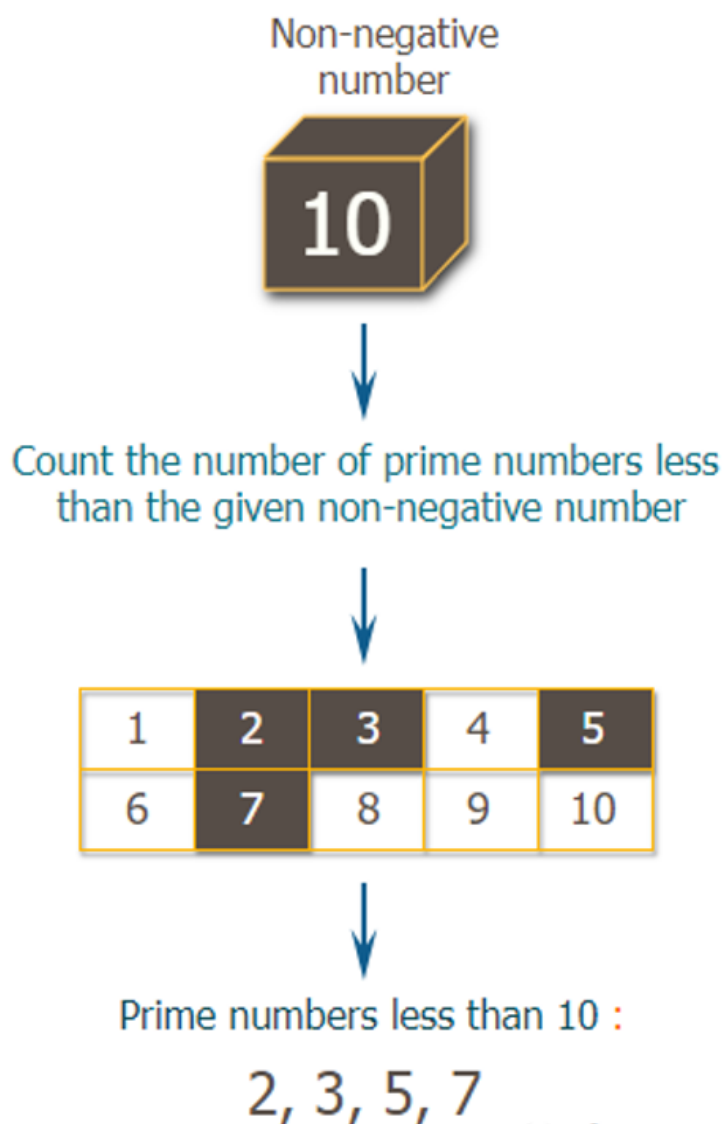
16

Input number of straight lines (o to exit):

0

20. Write a Python program to count the number of prime numbers less than a given non-negative number.

Pictorial Presentation:



```
In [35]: c=0
b = int(input("Enter stop:"))
for j in range(2,b+1):
    flag=0
    for i in range(2,j):
        if (j%i==0):
            flag=1
    if(flag==0):
        c+=1
print(c)
```

```
Enter stop:10
4
```

21. There are 12 intermediate stations between two places A and B. Write a Python program to find the number of ways in which a train can be made to stop at 4 of these intermediate stations so that no two stopping stations are consecutive?

Formula to find is :

$${}^nC_r = \frac{n!}{r!(n-r)!}$$

Where n=total number of non stop stations (Here n=12-4+1=9 non stop solutions) and r= no. of stops (Here r=4)



Examples –

Input : n = 12, s = 4

Output : 126

Input : n = 16, s = 5

Output : 792

```
In [41]: stations = int(input('Enter number of stations: '))
stops = int(input('Enter number of stops: '))\

n = stations - stops + 1
r = stops

#Computing n!
n_fact = 1
temp = n
while temp > 0:
    n_fact *= temp
    temp -= 1

#Computing r!
r_fact = 1
temp = r
while temp > 0:
    r_fact *= temp
    temp -= 1

#Computing (n-r)!
n_minus_r_fact = 1
temp = n - r
while temp > 0:
    n_minus_r_fact *= temp
    temp -= 1

#Computing nCr
ans = int(n_fact / (r_fact * (n_minus_r_fact)))
print(ans)
```

```
Enter number of stations: 12
Enter number of stops: 4
126
```

22. Python Program for Number of solutions to Modular Equations

Given A and B, the task is to find the number of possible values that X can take such that the given modular equation $(A \bmod X) = B$ holds good. Here, X is also called a solution of the modular equation.

Examples:

Input : A = 26, B = 2

Output : 6

Explanation

X can be equal to any of {3, 4, 6, 8, 12, 24} as A modulus any of these values equals 2 i. e., $(26 \bmod 3) = (26 \bmod 4) = (26 \bmod 6) = (26 \bmod 8) = \text{Output:} 2$

Input : 21 5

Output : 2

Explanation

X can be equal to any of {8, 16} as A modulus any of these values equals 5 i.e. $(21 \bmod 8) = (21 \bmod 16) = 5$

If we carefully analyze the equation $A \bmod X = B$ it is easy to note that if $(A = B)$ then there are infinitely many values greater than A that X can take. In the Case when $(A < B)$, there cannot be any possible value of X for which the modular equation holds. So the only case we are left to investigate is when $(A > B)$. So now we focus on this case in depth.

```
In [43]: a = int(input('Enter a: '))
          b = int(input('Enter b: '))

          if a == b:
              print('There are infinite solutions')
          elif a < b:
              print('Invalid')
          else:
              count = 0
              for i in range(b, a+1):
                  if a % i == b:
                      count += 1
              print(count)
```

Enter a: 21

Enter b: 5

2

23. Python program to check if the given number is a Disarium Number

A number is said to be the Disarium number when the sum of its digit raised to the power of their respective positions becomes equal to the number itself.

For example, 175 is a Disarium number as follows:

$$1^1 + 7^2 + 5^3 = 1 + 49 + 125 = 175$$

```
In [48]: n = int(input('Enter a number: '))

digits = len(str(n))
sum = 0
temp = n
for i in range(digits, 0, -1):
    d = n % 10
    sum += d**i
    n = n//10

if sum == temp:
    print(temp, 'is a Disarium number')
else:
    print(temp, 'is not a Disarium number')
```

```
Enter a number: 175
175 is a Disarium number
```

24. Python program to determine whether the given number is a Harshad Number.

If a number is divisible by the sum of its digits, then it will be known as a Harshad Number.

For example:

The number 156 is divisible by the sum (12) of its digits (1, 5, 6).

Some Harshad numbers are 8, 54, 120, etc.

In this program, integer value is predefined, we don't need to put integer value to determine whether the given number is a Harshad number or not

```
In [53]: num = int(input("Enter the number:"))
rem = sum = 0;

#Make a copy of num and store it in variable n
n = num;

#Calculates sum of digits
while(num > 0):
    rem = num%10;
    sum = sum + rem;
    num = num//10;

#Checks whether the number is divisible by the sum of digits
if(n%sum == 0):
    print(str(n) + " is a harshad number");
else:
    print(str(n) + " is not a harshad number");
```

Enter the number:156
156 is a harshad number

25. Python program to print all pronic numbers between 1 and 100

The pronic number is a product of two consecutive integers of the form: $n(n+1)$.

For example:

$$6 = 2(2+1) = n(n+1),$$

$$72 = 8(8+1) = n(n+1)$$

Some pronic numbers are: 0, 2, 6, 12, 20, 30, 42, 56 etc.

In this program, we need to print all pronic numbers between 1 and 100

```
In [54]: flag = 0
#Displays pronic numbers between 1 and 100
print("Pronic numbers between 1 and 100: ");
for i in range(1, 101):
    for j in range(1, i+1):
        #Checks for pronic number by multiplying consecutive numbers
        if((j*(j+1)) == i):
            flag = 1
            print(i,end=",")
            break
```

Pronic numbers between 1 and 100:
2,6,12,20,30,42,56,72,90,

26. Write a Python program to find an integer exponent x such that $a^x = n$.

```
Input:
a = 2 : n = 1024
Output:
10
Input:
a = 3 : n = 81
Output:
4
Input:
a = 3 : n =
12900700781701026662481960358450703949334417416449930858101164
41344597492642263849
Output:
170
```

```
In [57]: a=int(input("Enter the integer whose power needs to be calculated:"))
n=int(input("Enter the integer output:"))
m = 1
x = 0
while m != n:
    x += 1
    m *= a
print(x)

Enter the integer whose power needs to be calculated:3
Enter the integer output:12900700781701026662481960358450703949334417416449930858101
16441344597492642263849
170
```