

7th Nov Project in conditional statement in python

- If conditional statement
- Mark Even and Odd
- The Else Statement
- Odd or Even
- Greatest of Three
- Calculator
- Closest Number
- Factorial
- Check Prime
- Fibonacci Number
- Check Strong Number

If Conditional Statement

```
In [1]: j_happy = True  
s_happy = True  
  
in_party = j_happy and s_happy  
print(in_party)  
  
# Explanation:  
# Since both are happy, you can go to the party.
```

True

```
In [2]: j_happy = True  
s_happy = False  
  
in_party = j_happy and s_happy  
print(in_party)
```

```
# Explanation:  
# Since both are happy, you can go to the party.
```

False

```
In [3]: j_happy = False  
s_happy = True  
  
in_party = j_happy and s_happy  
print(in_party)
```

```
# Explanation:  
# Since both are happy, you can go to the party.
```

False

```
In [4]: j_happy = False  
s_happy = False  
  
in_party = j_happy and s_happy  
print(in_party)
```

```
# Explanation:  
# Since both are happy, you can go to the party.
```

False

```
In [5]: santosh_present = True  
teacher_present = True  
  
can_start_class = santosh_present and teacher_present  
print(can_start_class)  
  
# Explanation:  
# Since both Santosh and the teacher are present, class can start.
```

True

```
In [6]: santosh_present = False  
teacher_present = True
```

```
can_start_class = santosh_present and teacher_present
print(can_start_class)

# Explanation:
# Since both Santosh and the teacher are present, class can start.
```

False

```
In [7]: santosh_present = True
teacher_present = False

can_start_class = santosh_present and teacher_present
print(can_start_class)

# Explanation:
# Since both Santosh and the teacher are present, class can start.
```

False

```
In [8]: santosh_present = False
teacher_present = False

can_start_class = santosh_present and teacher_present
print(can_start_class)

# Explanation:
# Since both Santosh and the teacher are present, class can start.
```

False

```
In [9]: santosh_present = True
teacher_present = True

if santosh_present and teacher_present:
    print("Class can start.")
else:
    print("Class cannot start.")
```

Class can start.

```
In [10]: santosh_present = True
teacher_present = False
```

```
if santosh_present and teacher_present:  
    print("Class can start.")  
else:  
    print("Class cannot start.")
```

Class cannot start.

```
In [11]: santosh_present = False  
teacher_present = True  
  
if santosh_present and teacher_present:  
    print("Class can start.")  
else:  
    print("Class cannot start.")
```

Class cannot start.

Mark Even and Odd

```
In [14]: x = 4  
  
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Even

```
In [15]: x = 5  
  
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Odd

```
In [16]: x = 6
```

```
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Even

```
In [17]: x = 10  
  
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Even

```
In [18]: x = 7  
  
if x % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Odd

The Else Statement

```
In [19]: a = 10  
  
if a > 100:  
    print("Big Number")  
else:  
    print("Number")
```

Number

```
In [20]: a = 110  
  
if a > 100:  
    print("Big Number")
```

```
else:  
    print("Number")
```

Big Number

```
In [21]: a = 20  
  
if a > 200:  
    print("Big Number")  
else:  
    print("Number")
```

Number

```
In [22]: a = 240  
  
if a > 200:  
    print("Big Number")  
else:  
    print("Number")
```

Big Number

```
In [25]: a = 10000000  
  
if a > 1000000:  
    print("Celebrity")  
else:  
    print("Panoti")
```

Celebrity

```
In [27]: a = 11111  
  
if a > 1000000:  
    print("Celebrity")  
else:  
    print("Panoti")
```

Panoti

```
In [28]: a = 20000000
```

```
if a > 1000000:  
    print("Celebrity")  
else:  
    print("Panoti")
```

Celebrity

odd or Even

```
In [29]: n = 15  
  
if n % 2 == 0:  
    print("True") # Even number  
else:  
    print("False") # Odd number
```

False

```
In [30]: n = 10  
  
if n % 2 == 0:  
    print("True") # Even number  
else:  
    print("False") # Odd number
```

True

```
In [31]: n = 8  
  
if n % 2 == 0:  
    print("True") # Even number  
else:  
    print("False") # Odd number
```

True

```
In [32]: n = 17  
  
if n % 2 == 0:  
    print("True") # Even number
```

```
else:  
    print("False") # Odd number
```

False

Greatest Of Three

```
In [45]: a = 1  
b = 2  
c = 3  
numbers = [a, b, c, ]  
numbers.sort(reverse=True)  
  
print("Greatest three numbers are:", numbers[:3])
```

Greatest three numbers are: [3, 2, 1]

```
In [51]: a = 1  
b = 2  
c = 3  
  
numbers = [a, b, c, ]  
numbers.sort(reverse=True)  
  
print("Lowest three numbers are:", numbers[2])
```

Lowest three numbers are: 1

```
In [52]: a = 1  
b = 2  
c = 3  
  
numbers = [a, b, c, ]  
numbers.sort(reverse=True)  
  
print("Lowest three numbers are:", numbers[1])
```

Lowest three numbers are: 2

```
In [53]: a = 1  
b = 2  
c = 3  
  
numbers = [a, b, c]  
numbers.sort(reverse=False)  
  
print("Lowest three numbers are:", numbers[2])
```

Lowest three numbers are: 3

```
In [55]: a = 1  
b = 2  
c = 3  
  
numbers = [a, b, c]  
numbers.sort(reverse=False)  
  
print("Lowest three numbers are:", numbers[0])
```

Lowest three numbers are: 1

```
In [33]: a = 1  
b = 2  
c = 3  
d = 4  
  
if a > b and a>c:  
    print(a)  
elif b > a and b > c:  
    print(b)  
else:  
    print(c)
```

3

```
In [34]: a = 1  
b = 2  
c = 3
```

```
d = 4

if a > b and a > c and a > d:
    print(a)
elif b > a and b > c and b > d:
    print(b)
elif c > a and c > b and c > d:
    print(c)
else:
    print(d)
```

4

```
In [42]: a = 1
b = 2
c = 3
d = 4
e = 5

if a > b and a > c and a > d and a > e:
    print(a)
elif b > a and b > c and b > d and b > e:
    print(b)
elif c > a and c > b and c > d and c > e:
    print(c)
elif d > a and d > b and d > c and d > e:
    print(d)
else:
    print(e)
```

5

```
In [43]: a = 1
b = 2
c = 3
d = 4
e = 5

if a > b and a > c and a > d and a > e:
    print(a)
elif b > a and b > c and b > d and b > e:
```

```
    print(b)
elif c > a and c > b and c > d and c > e:
    print(c)
elif d > a and d > b and d > c and d > e:
    print(d)
else:
    print(e)
```

5

```
In [44]: a = 1
b = 2
c = 3
d = 4
e = 5
```

```
if a < b and a < c and a < d and a < e:
    print(a)
elif b < a and b < c and b < d and b < e:
    print(b)
elif c < a and c < b and c < d and c < e:
    print(c)
elif d < a and d < b and d < c and d < e:
    print(d)
else:
    print(e)
```

1

```
In [ ]: a = 1
b = 2
c = 3
d = 4
e = 5
```

```
if a < b and a < c and a < d and a < e:
    print(a)
elif b < a and b < c and b < d and b < e:
    print(b)
elif c < a and c < b and c < d and c < e:
    print(c)
```

```
elif d < a and d < b and d < c and d < e:  
    print(d)  
else:  
    print(e)
```

In [56]:

```
a = 2  
b = 2  
c = 5  
  
if a >= b and a >= c:  
    print(a)  
elif b >= a and b >= c:  
    print(b)  
else:  
    print(c)
```

5

In [57]:

```
a = 2  
b = 2  
c = 7  
  
if a >= b and a >= c:  
    print(a)  
elif b >= a and b >= c:  
    print(b)  
else:  
    print(c)
```

7

In [58]:

```
a = 5  
b = 7  
c = 5  
  
if a >= b and a >= c:  
    print(a)  
elif b >= a and b >= c:  
    print(b)  
else:  
    print(c)
```

7

```
In [59]: a = 8
b = 8
c = 80

if a >= b and a >= c:
    print(a)
elif b >= a and b >= c:
    print(b)
else:
    print(c)
```

80

Calculator

Given two numbers a and b. You need to perform basic mathematical operations on them. You will be provided an integer named as operator.

If the operator equals to 1 add a and b, then print the result. If the operator equals to 2 subtract b from a, then print the result. If the operator equals to 3 multiply a and b, then print the result. If the operator equals to any other number, print "Invalid Input"(without quotes).

```
In [60]: a = 1
b = 2
operator = 3

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
```

```
    print(a / b)
else:
    print("Invalid operator")
```

2

```
In [93]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(2,1,3)
```

2

```
In [94]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(2,3,4)
```

```
0.6666666666666666
```

```
In [97]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(10,5,4)
```

```
2.0
```

```
In [61]: a = 4
b = 8
operator = 3

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")
```

```
32
```

```
In [62]: a = 46
b = 50
operator = 3
```

```
if operator == 1:  
    print(a + b)  
elif operator == 2:  
    print(a - b)  
elif operator == 3:  
    print(a * b)  
elif operator == 4:  
    print(a / b)  
else:  
    print("Invalid operator")
```

2300

```
In [63]: a = 4  
b = 8  
operator = 1  
  
if operator == 1:  
    print(a + b)  
elif operator == 2:  
    print(a - b)  
elif operator == 3:  
    print(a * b)  
elif operator == 4:  
    print(a / b)  
else:  
    print("Invalid operator")
```

12

```
In [64]: a = 4  
b = 8  
operator = 2  
  
if operator == 1:  
    print(a + b)  
elif operator == 2:  
    print(a - b)  
elif operator == 3:  
    print(a * b)  
elif operator == 4:
```

```
    print(a / b)
else:
    print("Invalid operator")
```

-4

```
In [65]: a = 4
b = 8
operator = 3

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")
```

32

```
In [66]: a = 4
b = 8
operator = 4

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")
```

0.5

```
In [67]: a = 2
b = 2
```

```
operator = 2

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")
```

0

In [68]:

```
a = 4
b = 8
operator = 5

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
    print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")
```

Invalid operator

In [69]:

```
a = 10
b = 10
operator = 2

if operator == 1:
    print(a + b)
elif operator == 2:
    print(a - b)
elif operator == 3:
```

```

        print(a * b)
elif operator == 4:
    print(a / b)
else:
    print("Invalid operator")

```

0

Closest Number

Given two integers n and m ($m \neq 0$). The problem is to find the number closest to n and divisible by m. If there is more than one such number, then output the one having the maximum absolute value

In [70]:

```

# Input
n = 13
m = 4

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer
if abs(n - num1) < abs(n - num2):
    print(num1)
else:
    print(num2)

```

12

In [75]:

```

# Input
n = 15
m = 5

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer
if abs(n - num1) < abs(n - num2):

```

```
    print(num1)
else:
    print(num2)
```

15

```
In [81]: # Input
n = 96
m = 12

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer
if abs(n - num1) < abs(n - num2):
    print(num1)
else:
    print(num2)
```

96

```
In [83]: # Input
n = 14
m = 6

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer
if abs(n - num1) < abs(n - num2):
    print(num1)
else:
    print(num2)
```

12

```
In [85]: # Input
n = 18
m = 2
```

```

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer
if abs(n - num1) < abs(n - num2):
    print(num1)
else:
    print(num2)

```

18

In [86]:

```

# Input
n = -15
m = 6

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer, and if tie choose one with greater absolute value
if abs(n - num1) < abs(n - num2):
    print(num1)
elif abs(n - num1) > abs(n - num2):
    print(num2)
else:
    # If both are equally close, choose the one with greater absolute value
    print(num1 if abs(num1) > abs(num2) else num2)

```

-18

In [87]:

```

# Input
n = -25
m = 6

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer, and if tie choose one with greater absolute value
if abs(n - num1) < abs(n - num2):

```

```

    print(num1)
elif abs(n - num1) > abs(n - num2):
    print(num2)
else:
    # If both are equally close, choose the one with greater absolute value
    print(num1 if abs(num1) > abs(num2) else num2)

```

-24

```
In [88]: # Input
n = -30
m = 5

q = n // m
num1 = m * q
num2 = m * (q + 1)

# Check which one is closer, and if tie choose one with greater absolute value
if abs(n - num1) < abs(n - num2):
    print(num1)
elif abs(n - num1) > abs(n - num2):
    print(num2)
else:
    # If both are equally close, choose the one with greater absolute value
    print(num1 if abs(num1) > abs(num2) else num2)
```

-30

```
In [98]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")
```

```
obj = Solution()
obj.calculate(2,1,3)
```

2

```
In [99]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(10,60,3)
```

600

```
In [100...]: class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(40,60,4)
```

0.6666666666666666

In [101...]

```
class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(40,50,2)
```

-10

In [102...]

```
class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
            print(a - b)
        elif operator == 3:
            print(a * b)
        elif operator == 4:
            print(a / b)
        else:
            print("Invalid operator")

obj = Solution()
obj.calculate(50,40,2)
```

10

In [103...]

```
class Solution:
    def calculate(self, a: int, b: int, operator: int) -> None:
        if operator == 1:
            print(a + b)
        elif operator == 2:
```

```
        print(a - b)
    elif operator == 3:
        print(a * b)
    elif operator == 4:
        print(a / b)
    else:
        print("Invalid operator")

obj = Solution()
obj.calculate(40,50,1)
```

90

```
In [117...]  
class Solution:  
    def closestNumber(self, n, m):  
        q = n // m  
        num1 = m * q  
        num2 = m * (q + 1)  
  
        if abs(n - num1) < abs(n - num2):  
            return num1  
        else:  
            return num2  
obj = Solution()  
print(obj.closestNumber(20,5))
```

20

```
In [105...]  
obj = Solution()  
print(obj.closestNumber(18, 2))
```

18

```
In [106...]  
obj = Solution()  
print(obj.closestNumber(-15, 2))
```

-14

```
In [108...]  
obj = Solution()  
print(obj.closestNumber(10, 3))
```

9

```
In [111...]: obj = Solution()
print(obj.closestNumber(20, 4))
```

20

Factorial

Given an integer n, write a program to return the factorial of n. The Factorial of a number is the product of all the numbers from 1 to n.

Note: 0 factorial is equal to 1.

```
In [118...]: class Solution:
    def factorial(self, n):
        result = 1
        for i in range(1, n + 1):
            result *= i
        return result
```

```
In [119...]: 1*2*6*5*8
```

```
Out[119...]: 480
```

```
In [1]: n = 10
fact = 1

for i in range(1, n + 1):
    fact = fact * i

print(fact)
```

3628800

```
In [2]: n = 20
fact = 1

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

```
print(fact)
```

2432902008176640000

```
In [3]: n = 101
         fact = 1

         for i in range(1, n + 1)
             fact = fact * i

         print(fact)
```

```
In [4]: n = 5  
       fact = 1  
  
       for i in range(1, n + 1)  
           fact = fact * i  
  
       print(fact)
```

120

```
In [5]: n = int(input())
fact = 1
for i in range(1, n + 1):
    fact = fact * i
print(fact)
```

3628800

```
In [6]: n = int(input(20))

fact = 1
for i in range(1, n + 1)
    fact = fact * i
```

```
print(fact)
```

479001600

Check Prime

Given an integer n check if n is prime or not. A prime number is a number that is divisible by 1 and itself only.

Note: Print "True" if n is prime, otherwise print "False".

```
In [7]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
    else:
        print(True)
else:
    print(False)
```

True

```
In [8]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
    else:
        print(True)
else:
    print(False)
```

False

```
In [9]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [10]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [11]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [12]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [13]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

True

```
In [15]: n = int(input())

if n > 1:
    for i in range(2, n):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [16]: n = int(input())

if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

True

```
In [17]: n = int(input())

if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [18]: n = int(input())

if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

True

```
In [19]: n = int(input())

if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

False

```
In [20]: n = int(input())

if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

True

```
In [21]: n = int(input())

# Your code here
if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
        else:
            print(True)
else:
    print(False)
```

True

```
In [22]: n = int(input())

# Your code here
if n > 1:
    for i in range(2, int(n**0.5) + 1):
        if n % i == 0:
            print(False)
            break
    else:
        print(True)
else:
    print(False)
```

False

Fibonacci Number

Given an integer n. Write a program to find the nth Fibonacci number.

$F(0)= 0, F(1)=1$ The nth Fibonacci number is given by the formula $F(n) = F(n-1) + F(n-2)$. The first few fibonacci numbers are: 0 1 1 2 3 5. . .

```
In [23]: n = int(input())

# Your code here
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

print(b)
```

3

```
In [24]: n = int(input())

# Your code here
```

```
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

print(b)
```

8

In [25]:

```
n = int(input())

# Your code here
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

print(b)
```

5

In [26]:

```
n = int(input())

# Your code here
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

print(b)
```

1

In [27]:

```
n = int(input())

# Your code here
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
```

```
a = b  
b = c  
  
print(b)
```

55

In [28]:

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

```
# Your code here  
a, b = 0, 1  
for i in range(2, n + 1):  
    c = a + b  
    a = b  
    b = c
```

```
print(b)
```

13

In [29]:

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

```
# Your code here  
a, b = 0, 1  
for i in range(2, n + 1):  
    c = a + b  
    a = b  
    b = c
```

```
print(b)
```

10946

In [30]:

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

```
# Your code here  
a, b = 0, 1  
for i in range(2, n + 1):  
    c = a + b  
    a = b  
    b = c
```

```
print(b)
```

354224848179261915075

```
In [31]: # Back-end complete function Template for Python 3
n = int(input())

##### Write your code below #####
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

fib = b
##### Write your code above #####
print(fib)
```

21

```
In [32]: # Back-end complete function Template for Python 3
n = int(input())

##### Write your code below #####
a, b = 0, 1
for i in range(2, n + 1):
    c = a + b
    a = b
    b = c

fib = b
##### Write your code above #####
print(fib)
```

10946

Strong Number

Strong Numbers are the numbers whose sum of factorial of digits is equal to the original number. Given a number N, the task is to check if it is a Strong Number or not. Print 1 if the Number is Strong, else Print 0.

```
In [33]: N = int(input())
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0

while temp > 0:
    digit = temp % 10
    sum_fact += factorial(digit)
    temp //= 10

if sum_fact == N:
    print(1)
else:
    print(0)
```

1

```
In [34]: N = int(input())
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0

while temp > 0:
    digit = temp % 10
    sum_fact += factorial(digit)
    temp //= 10
```

```
if sum_fact == N:  
    print(1)  
else:  
    print(0)
```

1

```
In [35]: N = int(input())  
def factorial(num):  
    fact = 1  
    for i in range(1, num + 1):  
        fact *= i  
    return fact  
  
temp = N  
sum_fact = 0  
  
while temp > 0:  
    digit = temp % 10  
    sum_fact += factorial(digit)  
    temp //= 10  
  
if sum_fact == N:  
    print(1)  
else:  
    print(0)
```

1

```
In [36]: N = int(input())  
def factorial(num):  
    fact = 1  
    for i in range(1, num + 1):  
        fact *= i  
    return fact  
  
temp = N  
sum_fact = 0  
  
while temp > 0:  
    digit = temp % 10
```

```
    sum_fact += factorial(digit)
    temp //= 10

if sum_fact == N:
    print(1)
else:
    print(0)
```

1

```
In [37]: N = int(input())
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0

while temp > 0:
    digit = temp % 10
    sum_fact += factorial(digit)
    temp //= 10

if sum_fact == N:
    print(1)
else:
    print(0)
```

0

```
In [38]: N = int(input())
def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0
```

```
while temp > 0:
    digit = temp % 10
    sum_fact += factorial(digit)
    temp //= 10

if sum_fact == N:
    print("YES")
else:
    print("NO")
```

YES

```
In [44]: N = int(input())

def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0
exp_str = []

while temp > 0:
    digit = temp % 10
    fact_val = factorial(digit)
    sum_fact += fact_val
    exp_str.append(f"{digit}! = {fact_val}")
    temp //= 10

explanation = " + ".join(exp_str[::-1]) + f" = {sum_fact}"

if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✓")
else:
    print("NO")
```

```
print(f"Explanation: {explanation}")
print(f"So, {N} is NOT a Strong Number ✗")
```

YES

Explanation: $1! = 1 + 4! = 24 + 5! = 120 = 145$
 So, 145 is a Strong Number ✓

```
In [45]: N = int(input())

def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0
exp_str = []

while temp > 0:
    digit = temp % 10
    fact_val = factorial(digit)
    sum_fact += fact_val
    exp_str.append(f"{digit}! = {fact_val}")
    temp //= 10

explanation = " + ".join(exp_str[::-1]) + f" = {sum_fact}"

if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✓")
else:
    print("NO")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is NOT a Strong Number ✗")
```

YES

Explanation: $1! = 1 = 1$
 So, 1 is a Strong Number ✓

```
In [46]: N = int(input())

def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0
exp_str = []

while temp > 0:
    digit = temp % 10
    fact_val = factorial(digit)
    sum_fact += fact_val
    exp_str.append(f"{digit}! = {fact_val}")
    temp //= 10

explanation = " " + ".join(exp_str[::-1]) + f" = {sum_fact}"

if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✅")
else:
    print("NO")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is NOT a Strong Number ❌")
```

NO

Explanation: $1! = 1 + 4! = 24 = 25$
 So, 14 is NOT a Strong Number ❌

```
In [47]: N = int(input())

def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
```

```

    return fact

temp = N
sum_fact = 0
exp_str = []

while temp > 0:
    digit = temp % 10
    fact_val = factorial(digit)
    sum_fact += fact_val
    exp_str.append(f"{digit}! = {fact_val}")
    temp //= 10

explanation = " + ".join(exp_str[:-1]) + f" = {sum_fact}"

if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✓")
else:
    print("NO")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is NOT a Strong Number ✗")

```

YES

Explanation: $4! = 24 + 0! = 1 + 5! = 120 + 8! = 40320 + 5! = 120 = 40585$

So, 40585 is a Strong Number ✓

In [48]: N = int(input())

```

def factorial(num):
    fact = 1
    for i in range(1, num + 1):
        fact *= i
    return fact

temp = N
sum_fact = 0
exp_str = []

while temp > 0:

```

```

digit = temp % 10
fact_val = factorial(digit)
sum_fact += fact_val
exp_str.append(f"{digit}! = {fact_val}")
temp //= 10

explanation = " + ".join(exp_str[::-1]) + f" = {sum_fact}"

if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✓")
else:
    print("NO")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is NOT a Strong Number ✗")

```

NO

Explanation: $1! = 1 + 2! = 2 + 3! = 6 = 9$

So, 123 is NOT a Strong Number ✗

In [59]:

```

class Solution:
    def isStrong(self, N):
        # Helper function to find factorial of a digit
        def factorial(num):
            fact = 1
            for i in range(1, num + 1):
                fact *= i
            return fact

        temp = N
        sum_fact = 0
        exp_str = [] # explanation ke liye list

        # Calculate factorial sum of digits
        while temp > 0:
            digit = temp % 10
            fact_val = factorial(digit)
            sum_fact += fact_val
            exp_str.append(f"{digit}! = {fact_val}")
            temp //= 10

```

```
# Explanation string ready
explanation = " + ".join(exp_str[::-1]) + f" = {sum_fact}"

# Print result with explanation
if sum_fact == N:
    print("YES")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is a Strong Number ✓")
else:
    print("NO")
    print(f"Explanation: {explanation}")
    print(f"So, {N} is NOT a Strong Number ✗")
```

```
In [60]: obj = Solution()
obj.isStrong(145)
```

YES
Explanation: $1! = 1 + 4! = 24 + 5! = 120 = 145$
So, 145 is a Strong Number ✓

```
In [61]: obj = Solution()
obj.isStrong(123)
```

NO
Explanation: $1! = 1 + 2! = 2 + 3! = 6 = 9$
So, 123 is NOT a Strong Number ✗

```
In [62]: obj = Solution()
obj.isStrong(2)
```

YES
Explanation: $2! = 2 = 2$
So, 2 is a Strong Number ✓

```
In [63]: obj = Solution()
obj.isStrong(40858)
```

NO
Explanation: $4! = 24 + 0! = 1 + 8! = 40320 + 5! = 120 + 8! = 40320 = 80785$
So, 40858 is NOT a Strong Number ✗

```
In [64]: obj = Solution()
obj.isStrong(40585)
```

YES

Explanation: $4! = 24 + 0! = 1 + 5! = 120 + 8! = 40320 + 5! = 120 = 40585$
So, 40585 is a Strong Number ✓

```
In [65]: obj = Solution()
obj.isStrong(1432)
```

NO

Explanation: $1! = 1 + 4! = 24 + 3! = 6 + 2! = 2 = 33$
So, 1432 is NOT a Strong Number ✗

```
In [66]: obj = Solution()
obj.isStrong(1435)
```

NO

Explanation: $1! = 1 + 4! = 24 + 3! = 6 + 5! = 120 = 151$
So, 1435 is NOT a Strong Number ✗

```
In [67]: obj = Solution()
obj.isStrong(143)
```

NO

Explanation: $1! = 1 + 4! = 24 + 3! = 6 = 31$
So, 143 is NOT a Strong Number ✗

```
In [74]: obj = Solution()
obj.isStrong(1)
```

YES

Explanation: $1! = 1 = 1$
So, 1 is a Strong Number ✓

```
In [ ]:
```