

## ✓ This notebook contains the various examples for the conditional and loop statements

This notebook can be accessed via the Github Link ([https://github.com/amitvsuryavanshi04/SIC\\_programming\\_and\\_coding](https://github.com/amitvsuryavanshi04/SIC_programming_and_coding))

Program 11 program to check if a number is positive, negative or zero

```
num = float(input("Enter a number to be check +/-/0: "))
if num > 0 :
    print("Positive number")
elif num == 0 :
    print("Zero")
else :
    print("Negative number")
```

➡ Enter a number to be check +/-/0: 20  
Positive number

Program 12 to check if a number is odd or even

```
num = int(input("Enter a number to be check odd/even: "))
if (num % 2) == 0 :
    print("{0} is Even".format(num))
else :
    print("{0} is Odd".format(num))
```

➡ Enter a number to be check odd/even: 20  
20 is Even

Program 13 to check a year is leap or not

```
year = int(input("Enter a year: "))
if year % 4 == 0 and year % 100 != 0 or year % 400 == 0:
```

```

    print(f"{year} is a leap year.")
else:
    print(f"{year} is not a leap year.")

```

↩ Enter a year: 2025  
2025 is not a leap year.

Program 14 program to check number is prime or not,

$$p \in \mathbb{N}, p > 1, \text{ and } \forall d \in \mathbb{N}, (d \mid p) \Rightarrow (d = 1 \text{ or } d = p)$$

```

num = int(input("Enter a number: "))
# define a flag variable
flag = False
if num == 1:
    print(f"{num}, is not a prime number")
elif num > 1:
    # check for factors
    for i in range(2, num):
        if (num % i) == 0:
            flag = True # if factor is found, set flag to True
    # break out of loop
    break
# check if flag is True
if flag:
    print(f"{num}, is not a prime number")
else:
    print(f"{num}, is a prime number")

```

↩ Enter a number: 263  
263, is a prime number

Program 15 number to print all prime numbers between a particular interval

```

low_value = 1
upper_value = 25

print("Prime numbers between {0} and {1} are :".format(low_value, upper_value))

```

```

for num in range(low_value, upper_value + 1):
    if num > 1:
        for i in range(2, num):
            if (num % i) == 0:
                break
        else:
            print(num)

```

```

➦ Prime numbers between 1 and 25 are :
2
3
5
7
11
13
17
19
23

```

Program 16 to find the factorial of a number

$$n! = \begin{cases} 1, & \text{if } n = 0 \\ n \cdot (n-1)!, & \text{if } n > 0 \end{cases}$$

```

num = int(input("Enter a number: "))
factorial = 1
if num < 0:
    print("Factirial does not exist for negative numbers")
elif num == 0:
    print("Factorial of 0 is 1")
else:
    for i in range(1, num+1):
        factorial = factorial*i
    print(f'The factorial of {num} is {factorial}')

```

```

➦ Enter a number: 6
The factorial of 6 is 720

```

Program 17 to display a multiplication table

```
num = int(input("Display the Multiplication table of : "))

for i in range (1,11):
    print(f"{num} X {i} = {num*i}")
```

↔ Display the Multiplication table of : 12

```
12 X 1 = 12
12 X 2 = 24
12 X 3 = 36
12 X 4 = 48
12 X 5 = 60
12 X 6 = 72
12 X 7 = 84
12 X 8 = 96
12 X 9 = 108
12 X 10 = 120
```

Program 18 program to print fibonacci sequence

$$F_n = \begin{cases} 0, & \text{if } n = 0 \\ 1, & \text{if } n = 1 \\ F_{n-1} + F_{n-2}, & \text{if } n > 1 \end{cases}$$

```
nt = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0

if nt <= 0:
    print("!! Please Enter a positive integer !!")
elif nt == 1:
    print("Fibonacci sequence upto",nt,":")
    print(n1)
else:
    print("Fibonacci sequence:")
    while count < nt:
        print(n1)
        nth = n1 + n2
        n1 = n2
        n2 = nth
        count += 1
```

→ How many terms? 10  
 Fibonacci sequence:  
 0  
 1  
 1  
 2  
 3  
 5  
 8  
 13  
 21  
 34

Program 19 python program to check Armstrong number ?

$$\text{A number } N \text{ with } n \text{ digits is Armstrong} \iff N = \sum_{i=1}^n d_i^n$$

```
num = int(input("Enter a number: "))
# Calculate the number of digits in num
num_str = str(num)
num_digits = len(num_str)
# Initialize variables
sum_of_powers = 0
temp_num = num
# Calculate the sum of digits raised to the power of num_digits
while temp_num > 0:
    digit = temp_num % 10
    sum_of_powers += digit ** num_digits
    temp_num //= 10
# Check if it's an Armstrong number
if sum_of_powers == num:
    print(f"{num} is an Armstrong number.")
else:
    print(f"{num} is not an Armstrong number.")
```

→ Enter a number: 9474  
 9474 is an Armstrong number.

## Program 20 to print armstrong number in an interval

```
# Input the interval from the user
lower = int(input("Enter the lower limit of the interval: "))
upper = int(input("Enter the upper limit of the interval: "))
for num in range(lower, upper + 1): # Iterate through the numbers i
    order = len(str(num)) # Find the number of digits in 'num'
    temp_num = num
    sum = 0
    while temp_num > 0:
        digit = temp_num % 10
        sum += digit ** order
        temp_num //= 10
    # Check if 'num' is an Armstrong number
    if num == sum:
        print(num)
```

```
➡ Enter the lower limit of the interval: 10
Enter the upper limit of the interval: 1000
153
370
371
407
```