

| **TITLE: Decision Making Statements** |
| --- |

**AIM:** 1) Write a program to count the number of prime numbers and composite numbers entered by the user.

2) Write a program to check whether a given number is Armstrong or not.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:** Use different Decision Making statements in Python.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Resource Needed: Python IDE**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Theory:**

**Decision Control Statements**

**1) Selection/Conditional branching statements**

a) if statement

b) if-else statement

c) if-elif-else statement

**2)Basic loop Structures/Iterative statement**

a) while loop

b) for loop

**If statement:**

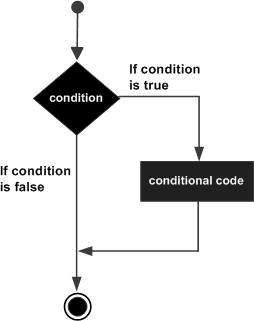
In Python **if** statement is used for decision-making operations. It contains a body of code which runs only when the condition given in the **if** statement is true.



Syntax:

if condition:

statement(s)

If flowchart:  


**If-else Statement:**

An **else** statement can be combined with an**if** statement. An **else** statement contains the block of code that executes if the conditional expression in the **if** statement resolves to 0 or a FALSE value.

The **else** statement is an optional statement and there could be at most only one **else**statement following **if**.

### Syntax:

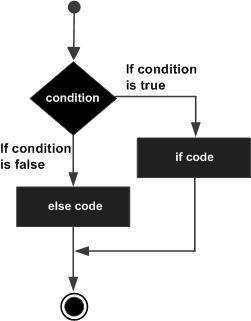
if expression:

statement(s)

else:

statement(s)

If-else flowchart:



## If-elif-else Statement:

The **elif** statement allows you to check multiple expressions for TRUE and execute a block of code as soon as one of the conditions evaluates to TRUE.

Similar to the else, the **elif** statement is optional. However, unlike **else**, for which there can be at most one statement, there can be an arbitrary number of **elif** statements following an **if.**



Syntax:

if expression1:

statement(s)

elif expression2:

statement(s)

elif expression3:

statement(s)

else:

statement(s)

**While loop:**

A **while** loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

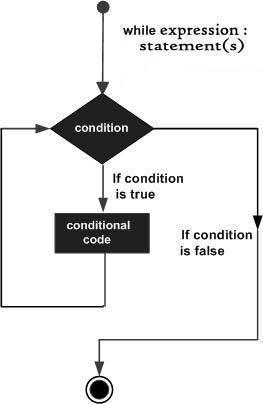


Syntax:

while expression:

statement(s)

While loop flowchart:



**For Loop:**

The [**for**](https://docs.python.org/3/reference/compound_stmts.html#for)statement in Python differs a bit from what you may be used to in C. Rather than giving the user the ability to define both the iteration step and halting condition (as C), Python’s **for**statement iterates over the items of any sequence (a list or a string), in the order that they appear in the sequence.

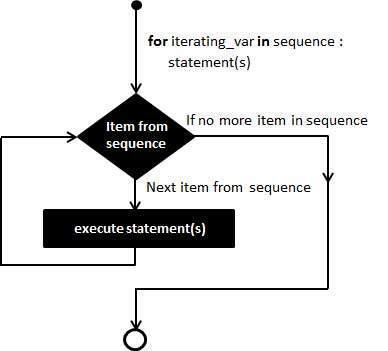


Syntax:

for iterating\_var in sequence:

statements(s)

For loop flowchart:



**Problem Definition:**

1)Write a program to read the numbers until -1 is encountered. Also, count the number of prime numbers and composite numbers entered by the user

2) Write a program to check whether a number is Armstrong or not.

## (Armstrong number is a number that is equal to the sum of cubes of its digits for example: 153 = 1^3 + 5^3 + 3^3.)

**Books/ Journals/ Websites referred:**

1. Reema Thareja, *Python Programming: Using Problem Solving Approach*, Oxford University Press, First Edition 2017, India
2. Sheetal Taneja and Naveen Kumar, *Python Programming: A modular Approach*, Pearson India, Second Edition 2018,India
3. https://docs.python.org/3/tutorial/controlflow.html#for-statements

**Implementation details:**

1)Write a program to read the numbers until -1 is encountered. Also, count the number of prime numbers and composite numbers entered by the user

**num = int(input("Enter Numbers:"))**

**x = 0**

**y = 0**

**while True:**

**if num == -1:**

**break**

**elif num == 1:**

**print(" is not a prime number")**

**elif num >1:**

**for i in range(2, num):**

**if num%i==0:**

**print(num, "is composite")**

**x+=1**

**print("Total number of composite numbers are :", x)**

**break**

**else:**

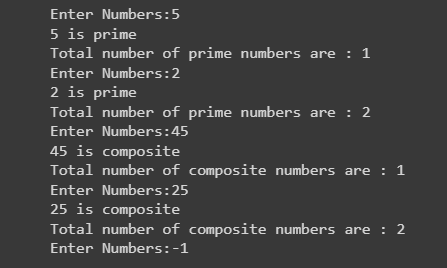
**print(num, "is prime")**

**y+=1**

**print("Total number of prime numbers are :", y)**

**num = int(input("Enter Numbers:"))**

Output :

****

2) Write a program to check whether a number is Armstrong or not.

## (Armstrong number is a number that is equal to the sum of cubes of its digits for example: 153 = 1^3 + 5^3 + 3^3.)

**a = int(input("Enter Number:"))**

**sum = 0**

**temp = a**

**while temp>0:**

**x = temp%10**

**sum += x\*\*3**

**temp //= 10**

**if a==sum:**

**print(a, "is an Armstrong Number")**

**else:**

**print(a, "is not an Armstrong Number")**

Output :

****

**Conclusion:**

Understood the use of different decision making statements in Python.

**Post Lab Questions:**

1) When should we use nested if statements? Illustrate your answer with the help of an example.

Nested if statements are used when you need to evaluate multiple conditions in a hierarchical or nested manner. They are typically used when you want to check a condition within another condition. Here's an example :

num = 15

if num >= 0:

if num == 0:

print("Zero")

else:

print("Positive number")

else:

print("Negative number"

In this example, nested if statements are used to determine a student's grade based on their numeric score. Each level of nesting checks a different condition, allowing for a more specific determination of the grade.

2) Explain the utility of break and continue statements with the help of an example.

break: It is used to exit the loop prematurely when a certain condition is met. For example, if you want to exit a loop when a specific value is found.

numbers = [1, 2, 3, 4, 5]

search\_value = 3

for num in numbers:

if num == search\_value:

print("Value found!")

break

continue: It is used to skip the rest of the current iteration and move to the next iteration of the loop. For example, if you want to skip printing even numbers:

numbers = [1, 2, 3, 4, 5]

for num in numbers:

if num % 2 == 0:

continue

print(num)

3) Write a program that accepts a string from user and calculate the number of digits and letters in string.

Here's a Python program that accepts a string from the user and calculates the number of digits and letters in the string.

input\_string = input("Enter a string: ")

letters = 0

digits = 0

for char in input\_string:

if char.isalpha():

letters += 1

elif char.isdigit():

digits += 1

print("Letters:", letters)

print("Digits:", digits)

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**