**Batch: C1\_3 Roll No.: 66**

**Experiment / assignment / tutorial No. 3**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **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

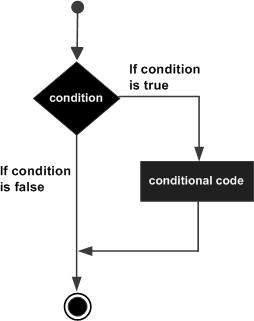
**If statement:**

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:

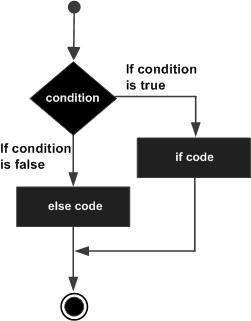
if expression:

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:**

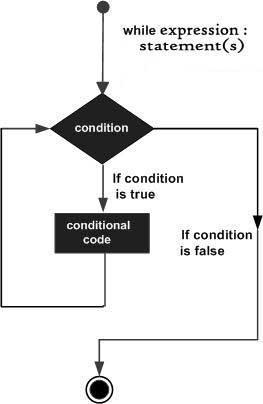
A **while** loop statement in Python programming language repeatedly executes a target statement as long as a given condition is true.

Syntax:

while expression:

statement(s)

While loop flowchart:



**For Loop:**

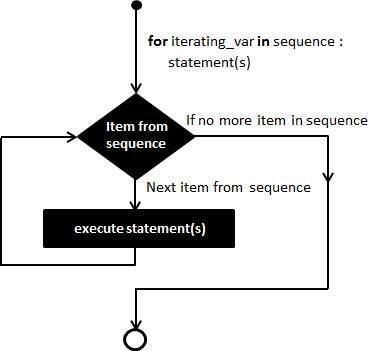
The [**for**](https://docs.python.org/3/reference/compound_stmts.html#for)statement in Python differs a bit from what you may be used to in C. Rather than giving the user the ability to define both the iteration step and halting condition (as C), Python’s **for**statement iterates over the items of any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

**Question 1:**

c=0

p=0

while(1):

  n=int(input());

  if(n==-1):

    break

  elif n==0 or n==1:

    continue

  else:

    for i in range(2,n):

      if(n%i==0):

        c+=1

        break;

    else:

          p+=1

print("No of composite nos. is : ",c);

print("No of prime nos. is :",p);

Question 2:

n=str(input())

copy=int(n)

l=[]

x=0

for i in n:

  i.split()

  l.append(i)

print(l)

for i in l:

  c=int(i)

  x+=(c\*\*3)

if(copy==x):

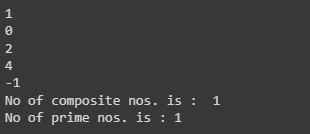
  print("Armsrtong")

else:

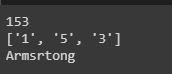
  print("U mom gae")

**Output(s):**

**Question 1:**



Question 2:



**Conclusion:**

Learn the use of different Decision Making statements and nested loops in Python.

**Post Lab Questions:**

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

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

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

Answers:

1)Nested if statements should be used when there are 2 or more conditions for a certain function to occur. For example if you have to print whether a given number is odd or even but only if the number is negative:

if(number<0):

if(number%2==0):

print(“even”)

else:

print(“odd”)

else:

print(“Number is not negative”)

##In this example one if statement checks whether the number is positive while the one nested in it checks whether the number is even or odd

2)

Break and continue are jump statements

Break : Ends and exits the loop or skips the code in that block

**Continue: skips the current iteration and jumps to the next**

Example: #taking n as an array of integers

for i in range(1,n):

print(i)

if(i%5==0):

break;

else:

continue;

**##In this example ,the loop breaks if it meets a multiple of 5, otherwise it continues**

**3)**

n=str(input())

alpha=0

digit=0

for i in n:

  if(i.isalpha()):

    alpha+=1

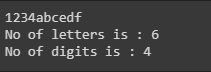
  if(i.isdigit()):

    digit+=1

print("No of letters is :",alpha)

print("No of digits is :",digit)

**Output:**



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