**Data Structures and Algorithm**

**Using Python**

**LAB FILE**

**SUBMITTED BY : SURAJ SAHU**

**ROLL NUMBER : 20232742**

**CLASS : B.VOC. 3RD SEMESTER**

**SUBMITTED TO: DR. RAHUL SOLANKI**



DEPARTMENT OF VOCATIONAL STUDIES

**RAMANUJAN COLLEGE,**

**UNIVERSITY**

**OF**

**DELHI**

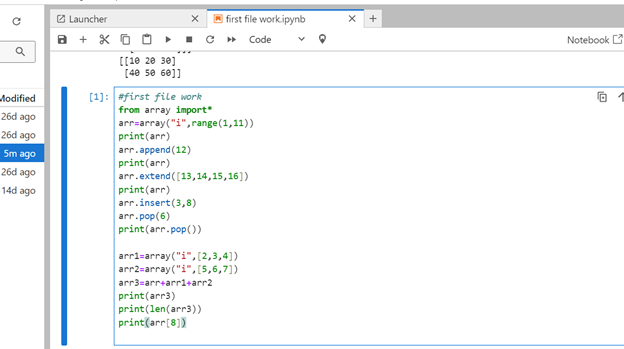
|  |  |
| --- | --- |
| **S.NO.** | **PRACTICAL** |
|  | Write a program in python to create an array and perform the following operations:   * 1. Add a new element to an array   2. Add four elements to an array   3. Add an element at the fourth position   4. Delete seventh element from the left of an array   5. Remove last element and return it   6. Create two arrays having three elements each and merge all three arrays.   7. Display the array formed in question   8. Determine the length of the new array formed   9. Print the ninth element of the new array |
| 2 | Write a program to create an array where user defines the length and input all its elements. |
| 3 | Write a code to create a 2D array and determine the following:   1. Datatype 2. Rank of the matrix 3. Order of the matrix 4. Convert 2D to 1D array 5. Create another 2x2 matrix and perform addition, subtraction and multiplication. |
| 4 | Write a python code to create a 3x3 array and determine the following:   1. Transpose of matrix 2. Create another 3x3 matrix and perform matrix multiplication. |
| 5 | Write a program for the following in a Singly linked list.:   1. Traversing 2. Insertion: 3. At the beginning 4. At the end 5. At any position 6. Deletion: 7. At the beginning 8. At the end 9. At any position |
| 6 | Write a program for the following in a Singly linked list.:   * 1. Traversing   2. Insertion:      1. At the beginning      2. At the end      3. At any position   3. Deletion:      1. At the beginning      2. At the end      3. At any position |
| 7 | Write a program for the following in a Circular linked list.:   * 1. Traversing   2. Insertion:      1. At the beginning      2. At the end      3. At any position   3. Deletion:      1. At the beginning      2. At the end      3. At any position |
| 8 | Create and perform different operation on Queue by using list:   1. Display 2. Enqueue 3. Dequeue 4. Exit |
| 9 | Create and perform different operation on Queue by using singly linked list:   1. Display 2. Enqueue 3. Dequeue 4. Exit |
| 10 | Write an algorithm for insertion, deletion, and display of elements in a stack (using LIST), where user can select and run any function of his/her choice. |
| 11 | Write an algorithm for insertion, deletion, and display of elements in a stack (using LINKED LIST), where user can select and run any function of his/her choice. |
| 12 | Write a program using recursion in python to find factorial of any user defined number. |
| 13 | Write a program using recursion in python to find fibonacci of any user defined number. |

**EXPERIMENT 1**

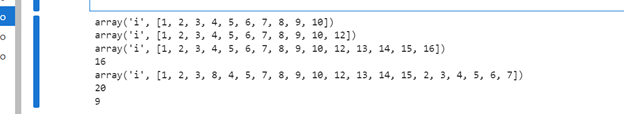
**AIM:** Create and perform different operations in an array.

**INTRODUCTION:** Definition of array,

**ALGORITHM:**



**OUTPUT:**

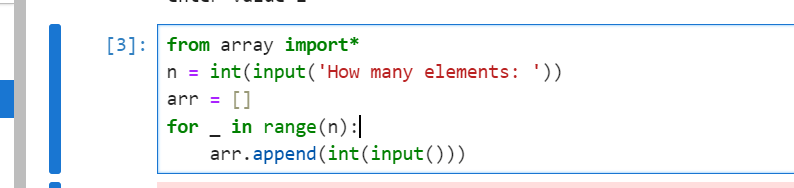


**EXPERIMENT 2**

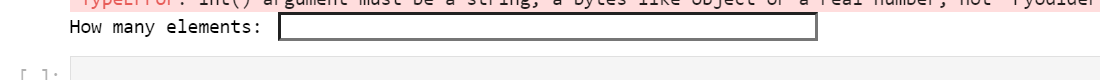
**AIM:** Create and perform different operations in an array.

**INTRODUCTION:** Definition of array and some operations,

**ALGORITHM:**



**OUTPUT:**

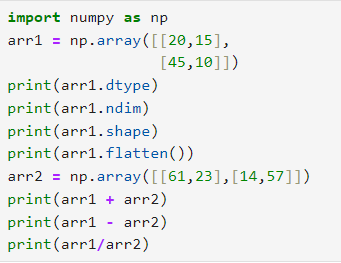


**EXPERIMENT 3**

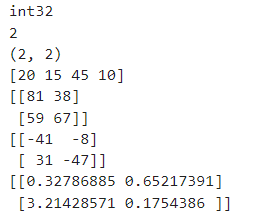
**AIM:** Program in python to create a 2x2 matrix and do different operations.

**INTRODUCTION:** Python can also be used on Matrix operations using built in module called numpy.

**ALGORITHM:**



**OUTPUT:**

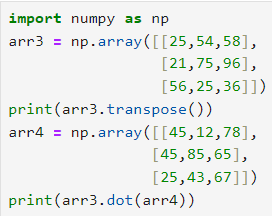


**EXPERIMENT 4**

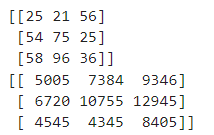
**AIM:** Python code to create a 3-D array using ‘Numpy’ and determine Transpose of a matrix and Create a 3-D array .

**INTRODUCTION:** Using python we can create 3d arrays also

**ALGORITHM:**



**OUTPUT:**

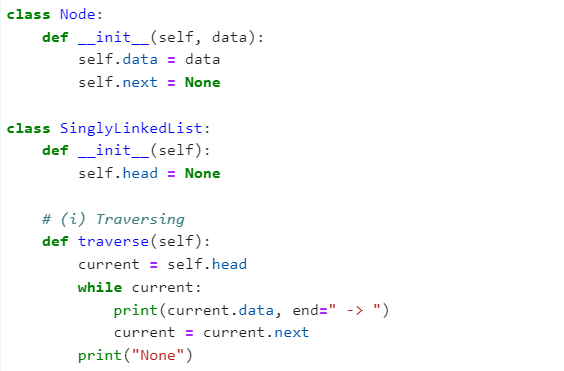


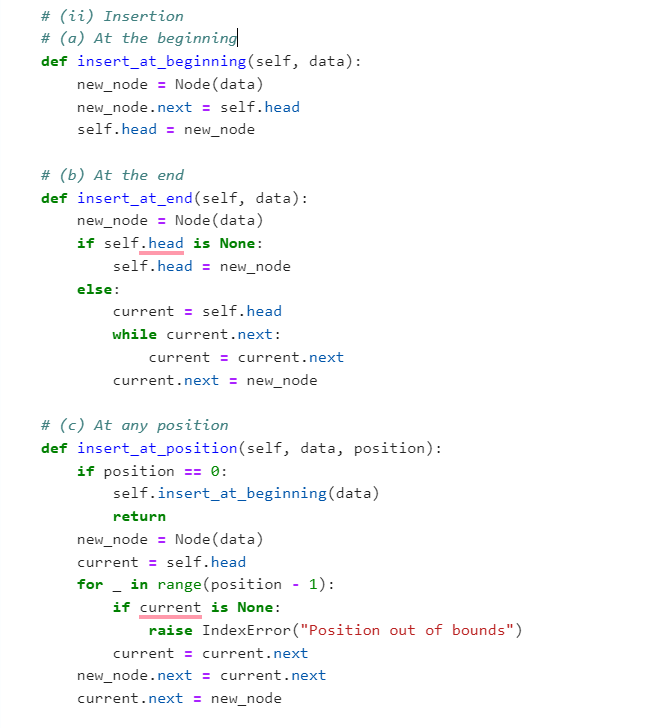
**EXPERIMENT 5**

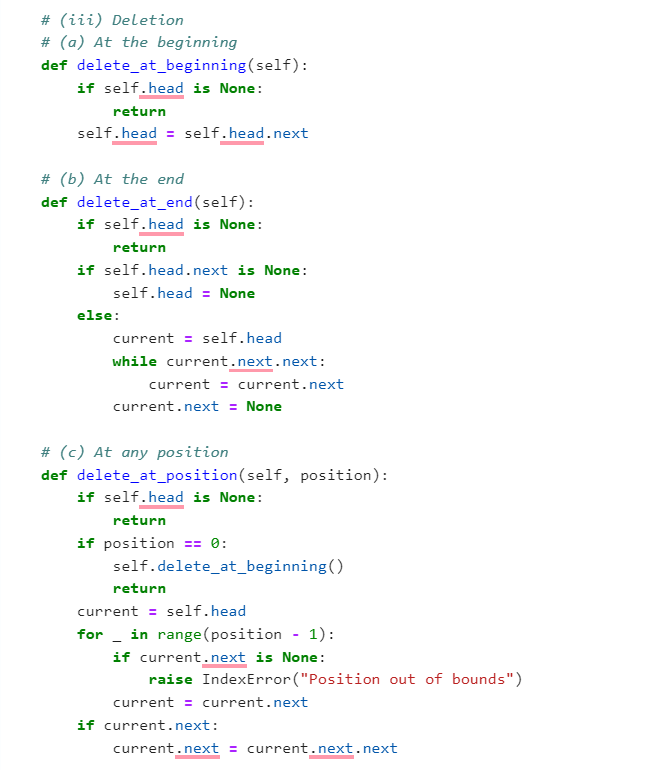
**AIM:** Python program to perform operations in a linked list.

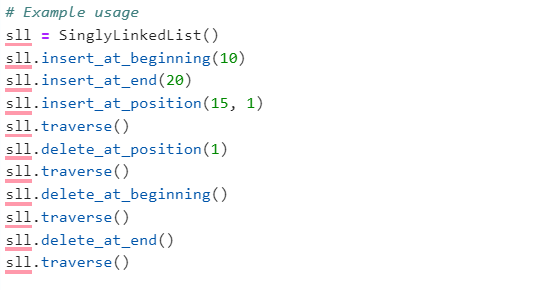
**INTRODUCTION:** A linked list is a connection of node containing data and reference.

**ALGORITHM:**

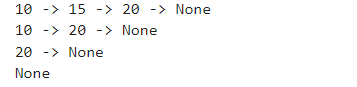








**OUTPUT:**

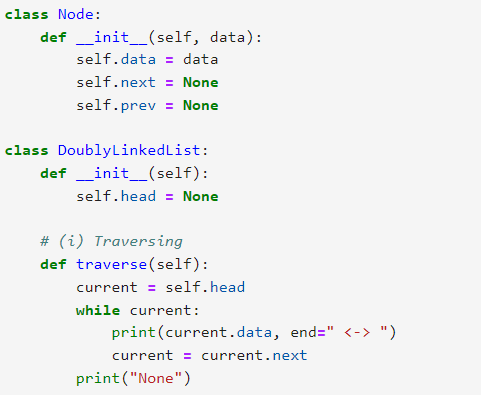


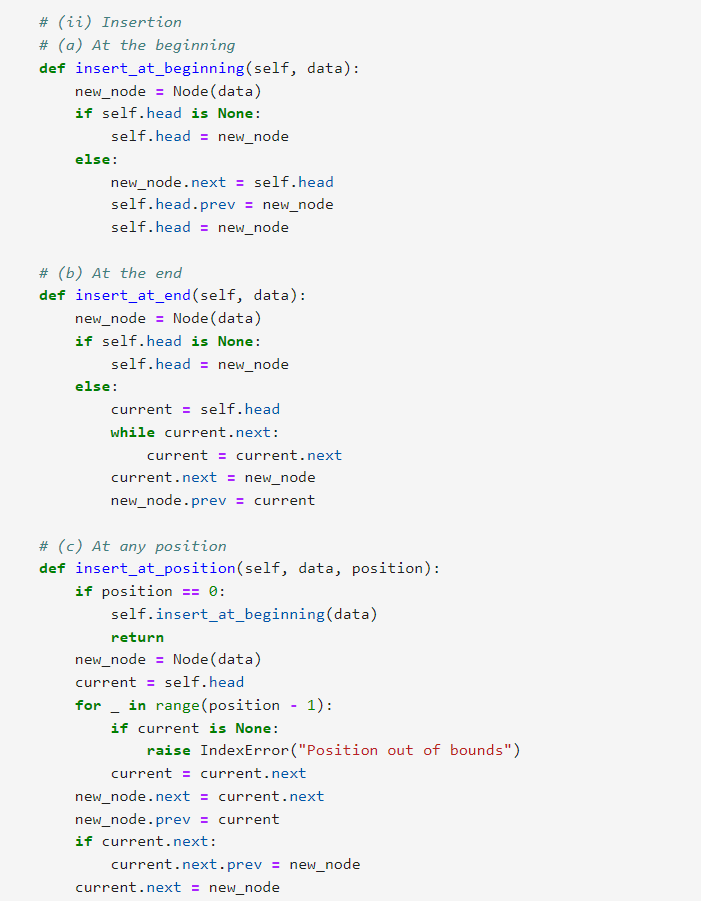
**EXPERIMENT 6**

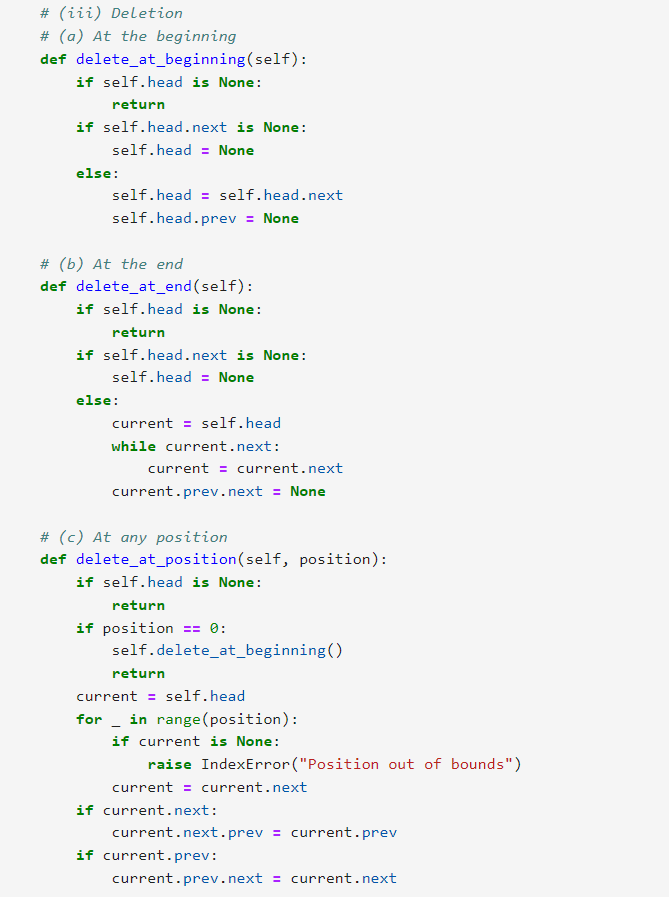
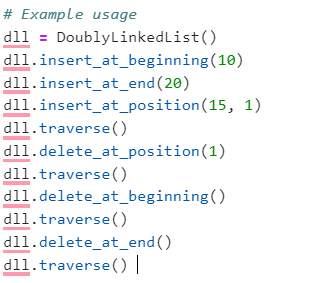
**AIM:** Python program to perform operations in a doubly linked list.

**INTRODUCTION:** Doubly linked list can be traversed in both directions.

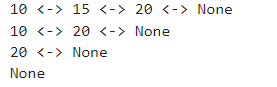
**ALGORITHM:**





**OUTPUT:**



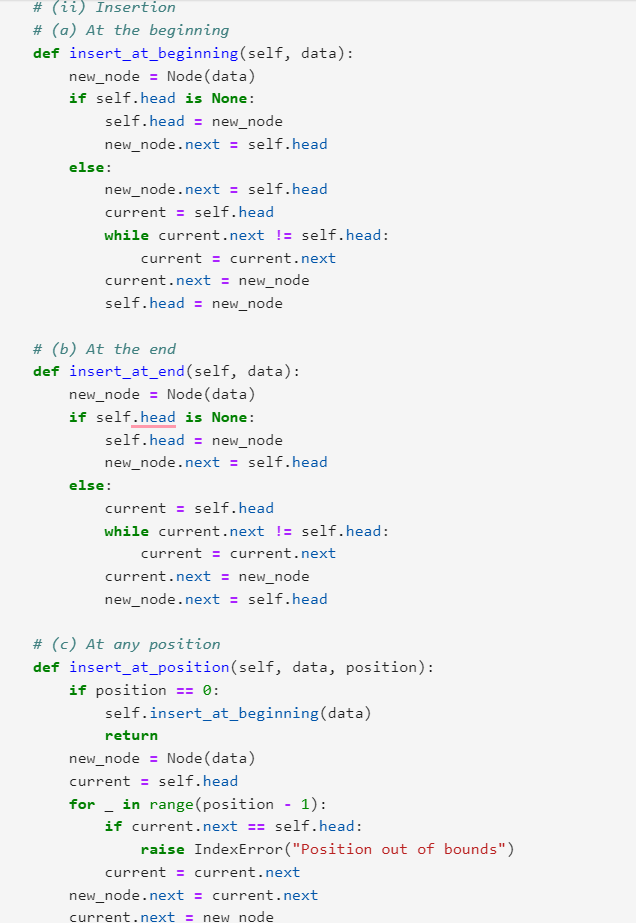
**EXPERIMENT 7**

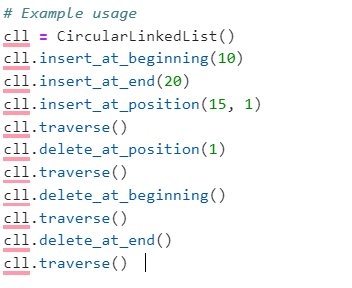
**AIM:** Python program to perform operations in a circular linked list.

**INTRODUCTION:** Ciircular linked list is in which the first node is connected to the last node.

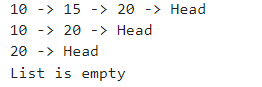
**ALGORITHM:**







**OUTPUT:**

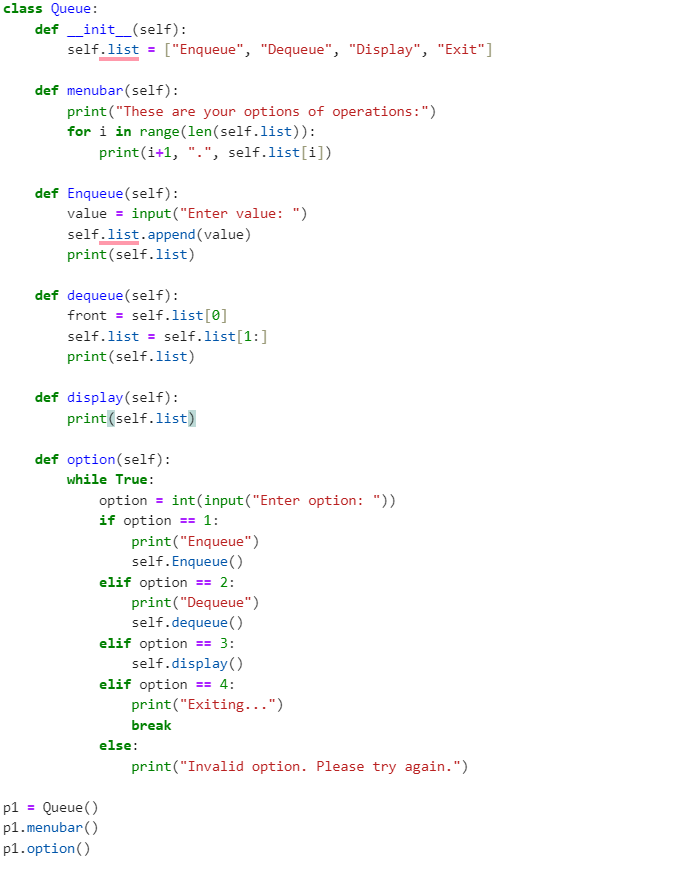


**EXPERIMENT 8**

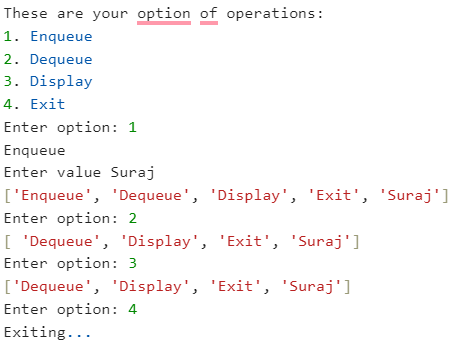
**AIM:** Python program to implement Queue Operations.

**INTRODUCTION:** Queue is a data structure with LIFO principle.

**ALGORITHM:**



**OUTPUT:**

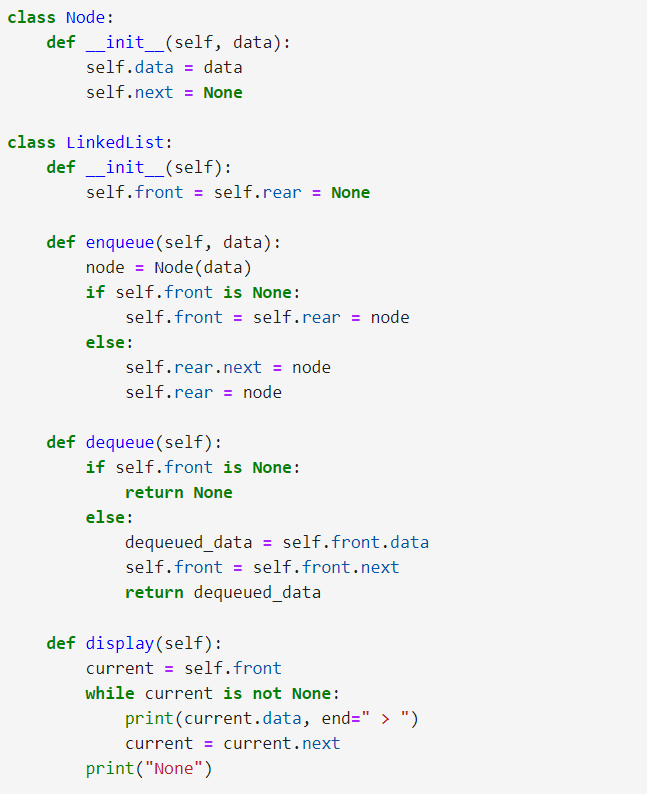


**EXPERIMENT 9**

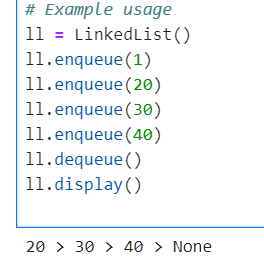
**AIM:** Python program to implement Queue Operations using singly linked list.

**INTRODUCTION:** Queue is a data structure with LIFO principle.

**ALGORITHM:**



**OUTPUT:**

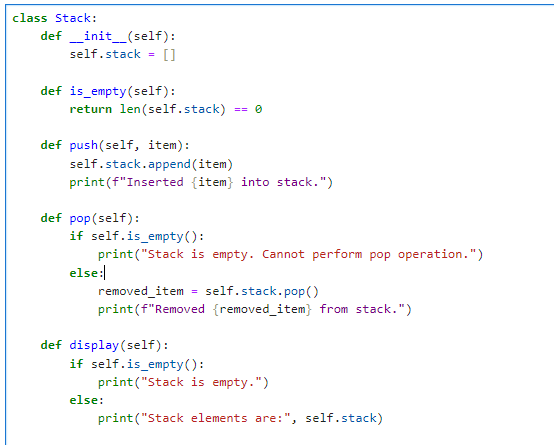


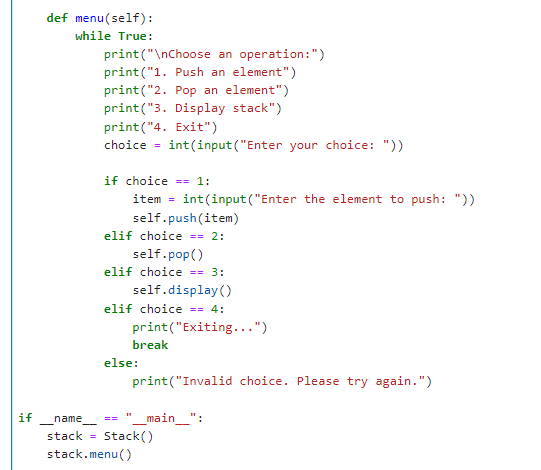
**EXPERIMENT 10**

**AIM:** Python program to implement Stack Operations using list.

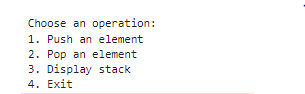
**INTRODUCTION:** Stack is a data structure with FILO/LOFI principle.

**ALGORITHM:**





**OUTPUT:**



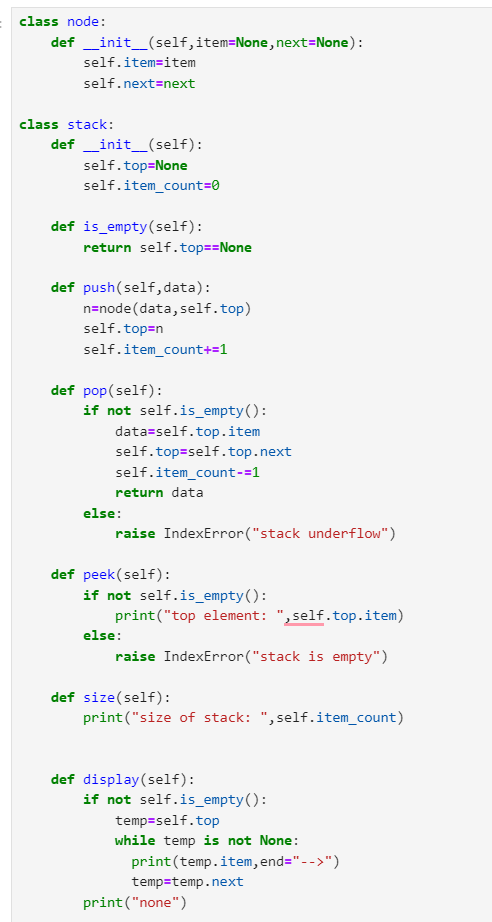


**EXPERIMENT 11**

**AIM:** Python program to implement Stack Operations using linked list.

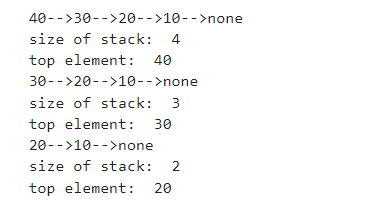
**INTRODUCTION:** Stack is a data structure with FILO/LOFI principle.

**ALGORITHM:**





**OUTPUT:**

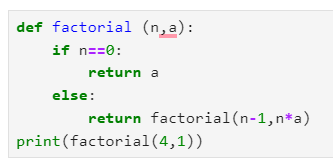


**EXPERIMENT 12**

**AIM:** Python program to find factorial of any number.

**INTRODUCTION:** Recursion is a process in which a function calls itself again and again.

**ALGORITHM:**



**OUTPUT:**

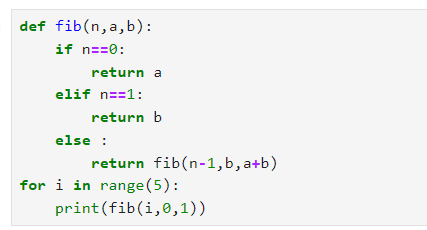


**EXPERIMENT 13**

**AIM:** Python program to find fibonacci of any number.

**INTRODUCTION:** Recursion is a process in which a function calls itself again and again.

**ALGORITHM:**



**OUTPUT:**

