***Introduction to Data Structure and Algorithm in PYTHON :* STACKS**

Activities for this lab:

- **Explain the concepts of Stack**
- ***Stack creation and implementation using Array and LL***
- ***lab exercise***

# STACKS

**OVERVIEW**
**A stack is a last in, first out (LIFO) data structure. Items are removed from a stack in the reverse order from the way they were inserted**

**AIM:**

- To perform all stacks operation:
  - append(object): inserts an element end of the list
  - insert(position, item): inserts an element in '*i*' th position in the list
  - pop(): removes and returns the last inserted element
  - pop(*i*): removes the '*i*' th positioned elements from list
  - len(itemName): returns the number of elements stored
  - boolean len(itemName) == 0: indicates whether no elements are stored
  - Use both Array and Linked list for implementation

ALGORITHM Steps:

**For Array Based,** Declare and initialize necessary variables, eg size.

Initialy **an empty list** and **myStack;**

1. Use **while** loop/or any control statement to get input from user.

2. Continue while loop until list size and defined size are same;

```
If len(list) == size
   print " stack overflow"
else
  Read item from user
```

```
        myStack = item
```

3. For next append operation, goto step 2.

4. For pop operation, checks that your defined size is equal to list size or not;

```
    If len(list) == size
      print "Stack underflow"
    Else
      myStack = pop();
      Display item
```

5. For next pop operation, goto step 4.
6. stop

**Lab Exercise (Individual)**

Modify the code given in the lab session, *stack.py,* to create a Menu that enable user to enter option for stack operations

1. append()
2. Pop()
3. Count last item of the list
4. len(list) == 0 and check whether list is full or not
5. len()
6. print()

Note:

Modify append() function so that it will enable you to push more items using (Y/N): E.g

Enter your Choice: 1
Push item to stack: 12
Enter more(y/n): y
Push item to stack: 15
Enter more(y/n): n

Modify Pop() function so that it will confirm user want to delete item or not,(give warning that item deleted cannot be recovered) using (Delete/Cancel): E.g (D/C)

**Example of INPUT / OUTPUT**

```
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit

1
Enter the number to be pushed: 11
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit
1
Enter the number to be pushed: 22
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit
1
Enter the number to be pushed: 33
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit
2
The number Popped is : 33
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit
3
The stack is 11 22
Enter the operation to be performed: 1) append  2) pop 3) print 4)
len 5) exit
4
The size of the stack is 2
```

All the stack operations are performed using the switch case (An Array based stack). Try LL based