KALYANI GOVERNMENT ENGINEERING COLLEGE

Department of Computer Application

Lab Assignment 5

Topic - Stack application & Linked list

Name: Madhusudan Chand
Roll No.: 10271023013
Registration No.: 231020510013
Stream: MCA
Semester: 2^{nd}

Paper: Date Structure Lab with python
Paper Code: MCAN-291

Contents

5.1 Write a Python program to convert an infix expression into postfix expression.

Ans -

Code

```
class Stack:
    def __init__(self,size):
        self.size=size
        self.top = -1
        self.a = []*size
    def __del__(self):
        print("Destruction")
    def push_s(self,x):

        if self.top < self.size -1:
            self.top+=1
            self.a.append(x)
        else:
            print("Overflow")
    def pop_s(self):
        if self.top != -1:

            val = self.a[self.top]
            self.a=self.a[:self.top]
            self.top-=1
            return val
        print("\nUnderflow")
    def display(self):
        k=self.top
        print("\nThe stack-->")
        while k!=-1:
            print(self.a[k],end=" ")
            k-=1
```

```
class Conversion:
    def __init__(self):
        pass
    def __del__(self):
        print("Destruction")
    def precedence(self,c):
        if c=='^':
            return 3
        elif c=='*' or c=='/' or c=='%':
            return 2
        elif c=='+' or c=='-':
            return 1
        else:
            return -1

    def isOperand(self,c):
        if '0'<=c<='9' or 'a'<=c<'z' or 'A'<=c<='Z':
            return True
        return False


    def associativity(self,c):
        if c == '^':
            return 'R'
        return 'L'

    def takeInfixex(self,s):
        st=Stack(len(s))
        result=""
        for i in s:
            if self.isOperand(i):
                result+=i
            elif i == '(':
                st.push_s(i)
            elif i==')':
```

```
                while st.a and st.a[st.top] != '(':
                    result+=str(st.pop_s())
                st.pop_s()
            else:
                while (st.a and (self.precedence(i) <
                    self.precedence(st.a[st.top])
                        or (self.precedence(i) ==
                        self.precedence(st.a[st.top])
                            and self.associativity(i) == 'L'))):
                    result+=st.pop_s()
                st.push_s(i)
        while st.a:
            result+=st.pop_s()
        print(result)
if __name__ =='__main__':

    c= Conversion()

    print("Equivalent postifix expression of 2+3-(5*5)")
    c.takeInfixex("(2+3)/(5*5)")
```

Output

```
Equivalent postifix expression of 2+3-(5*5)
23+55*/
Destruction
Destruction
```

5.2 Write a Python program to evaluate a postfix expression.

Ans -

Code:

```
import stackclas as st1
s = input("Enter your postfix expression\n")
st=st1.stackpro(len(s))
for i in s:
    if i.isdigit():
        st.push_s(i)
    else:
        x=st.pop_s()
        y=st.pop_s()
        st.push_s(str(eval(y + i + x)))
print(st.a[st.top])
```

Output

```
Enter your postfix expression
23-55*/

-0.04
Destroy
```

5.3 Write a C program to create a linked list to contain the vowels 'a', 'e', 'i', 'o', 'u' in the data field of the nodes.

Ans -

Code :

```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None

class LinkedList:
    def __init__(self):
        self.head=None
```

```
    def addNode(self,data):
        newNode = Node(data)
        if self.head == None:
            self.head = newNode
            return

        temp = self.head
        while temp.next:
            temp = temp.next
        temp.next=newNode
    def printList(self):
        if self.head == None:
            print("Empt linked list")
            return
        temp = self.head

        while temp:
            print(temp.data , end= " ")
            temp = temp.next

        print()
if __name__ == "__main__":
    print("Linked list containing vowels : ")
    l = LinkedList()
    l.addNode('a')
    l.addNode('e')
    l.addNode('i')
    l.addNode('o')
    l.addNode('u')

    l.printList()
```

Output

```
Linked list containing vowels :
a e i o u
```

5.4 Write a C program to delete the first node that contains an integer data item of a single linked list.

Ans -

Code:

```
class Node:
    def __init__(self,data):
        self.data=data
        self.next=None
class LinkedList:
    def __init__(self):
        self.head=None
    def addNode(self,data):
        newNode = Node(data)
        if self.head == None:
            self.head = newNode
            return

        temp = self.head
        while temp.next:
            temp = temp.next
        temp.next=newNode
    def deletefirst(self):
        if self.head == None:
            print("Linked list is empty")
            return
        if type(self.head.data) == int:
                temp = self.head
                print("Deleted item",temp.data)
                self.head=temp.next
        else:
```

```
                print("Not int")
    def printList(self):
        if self.head == None:
            print("Empt linked list")
            return
        temp = self.head
        while temp:
            print(temp.data , end= " ")
            temp = temp.next
        print()

if __name__ == "__main__":
    print("Linked list containing vowels : ")
    l = LinkedList()
    l.addNode(54)
    l.addNode('a')
    l.addNode('e')
    l.addNode('i')
    l.addNode('o')
    l.addNode('u')

    l.printList()
    l.deletefirst()
    l.printList()
```

Output

```
Linked list containing vowels :
54 a e i o u
Deleted item 54
a e i o u
```