Assignment 6

* Question 2

Implement push, pop and find the minimum element in a stack in O(1) time complexity.

Program-

class Node:

def \_\_init\_\_(self, value):

self.value = value

self.next = None

class Stack:

def \_\_init\_\_(self):

self.top = None

self.count = 0

self.minimum = None

def getMin(self):

if self.top is None:

return "Stack is empty"

else:

print("Minimum Element in the stack is: {}".format(self.minimum))

def push(self, value):

if self.top is None:

self.top = Node(value)

self.minimum = value

elif value < self.minimum:

temp = (2 \* value) - self.minimum

new\_node = Node(temp)

new\_node.next = self.top

self.top = new\_node

self.minimum = value

else:

new\_node = Node(value)

new\_node.next = self.top

self.top = new\_node

print("Number Inserted: {}".format(value))

def pop(self):

if self.top is None:

print("Stack is empty")

else:

removedNode = self.top.value

self.top = self.top.next

if removedNode < self.minimum:

print("Top Most Element Removed :{} ".format(self.minimum))

self.minimum = ((2 \* self.minimum) - removedNode)

else:

print("Top Most Element Removed : {}".format(removedNode))

stack = Stack()

stack.push(3)

stack.push(5)

stack.getMin()

stack.push(2)

stack.push(1)

stack.getMin()

stack.pop()

stack.getMin()

stack.pop()