







class Stack:  
 def push(self, e):  
 self.items.append(e)  
 def is\_empty(self):  
 return len(self.items) == 0  
 def top(self):  
 if self.is\_empty():  
 raise IndexError("top from empty stack")  
 return self.items[-1]  
 def pop(self):  
 if self.is\_empty():  
 raise IndexError("pop from empty stack")  
 return self.items.pop()  
 def \_\_init\_\_(self):  
 self.items = []  
 def \_\_len\_\_(self):  
 return len(self.items)  
  
# SIMULATION 1  
def Simulation1():  
 S = Stack()  
 results = []  
 S.push(5)  
 S.push(3)  
 results.append(len(S))  
 results.append(S.pop())  
 results.append(S.is\_empty())  
 results.append(S.pop())  
 results.append(S.is\_empty())  
 try:  
 results.append(S.pop())  
 except IndexError as e:  
 results.append(str(e))  
  
 S.push(7)  
 S.push(9)  
 results.append(S.top())  
 S.push(4)  
 results.append(len(S))  
 results.append(S.pop())  
 S.push(6)  
 S.push(8)  
 results.append(S.pop())  
 return results  
  
Simulation1\_answer = Simulation1()  
print("Simulation 1 Answers:")  
for result in Simulation1\_answer:  
 print(result)  
  
# SIMULATION 2  
def Simulation2():  
 X = Stack()  
 results = []  
 X.push(5)  
 X.push(3)  
 results.append(X.pop())  
 X.push(2)  
 X.push(8)  
 results.append(X.pop())  
 results.append(X.pop())  
 X.push(9)  
 X.push(1)  
 results.append(X.pop())  
 X.push(7)  
 X.push(6)  
 results.append(X.pop())  
 results.append(X.pop())  
 X.push(4)  
 results.append(X.pop())  
 results.append(X.pop())  
 return results  
  
Simulation2\_Answer = Simulation2()  
print()  
print("Simulation 2 Answers:")  
for result in Simulation2\_Answer:  
 print(result)