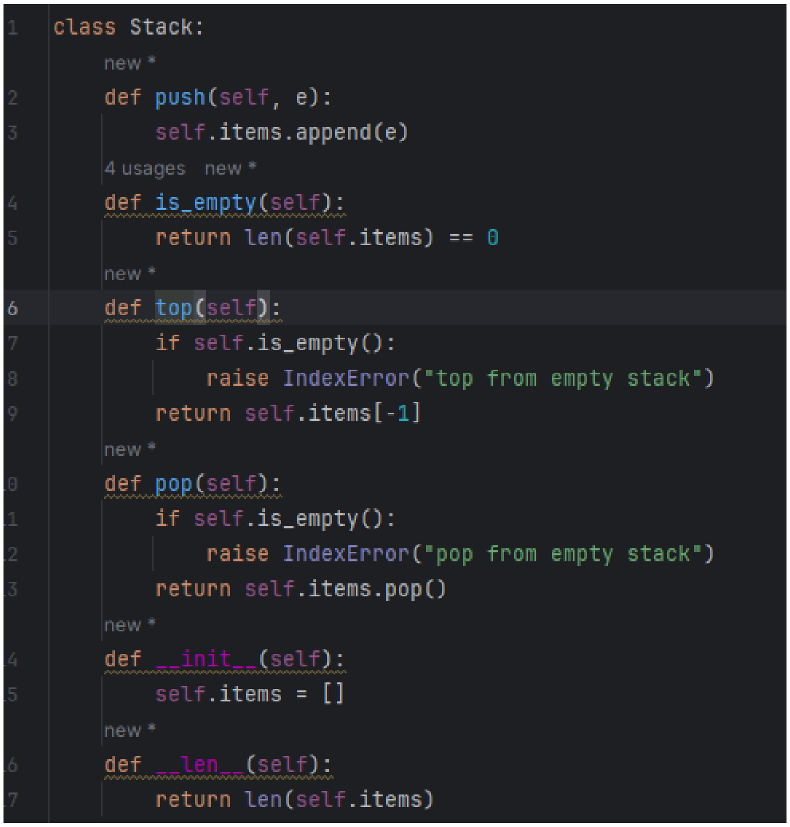
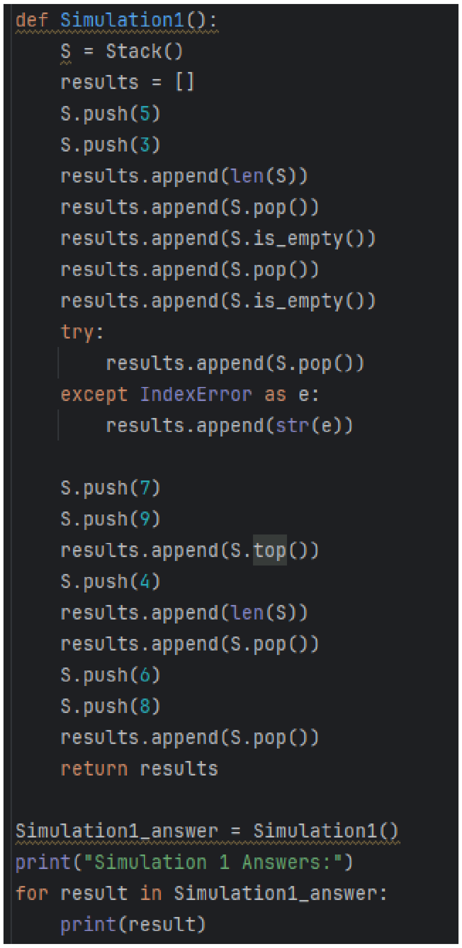
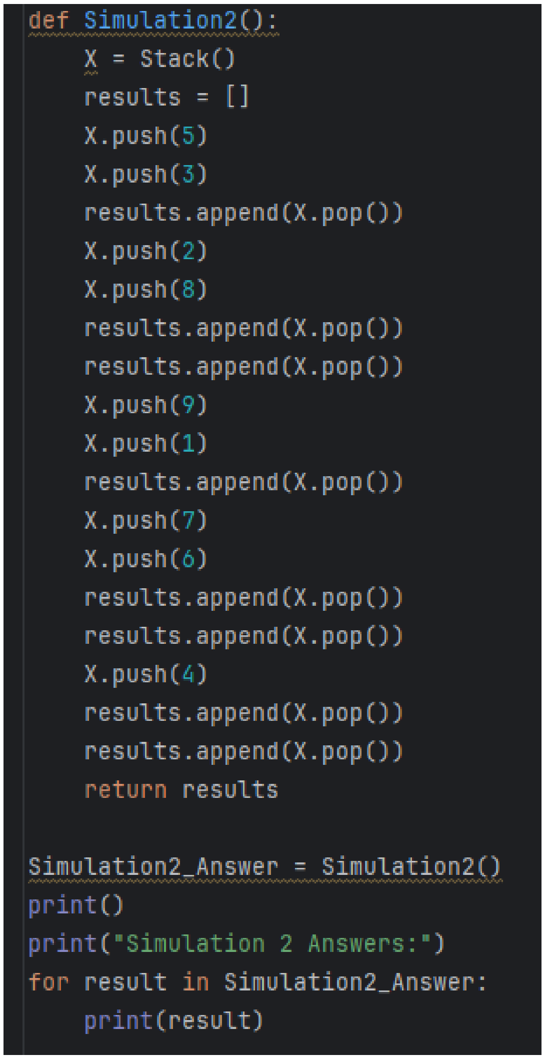


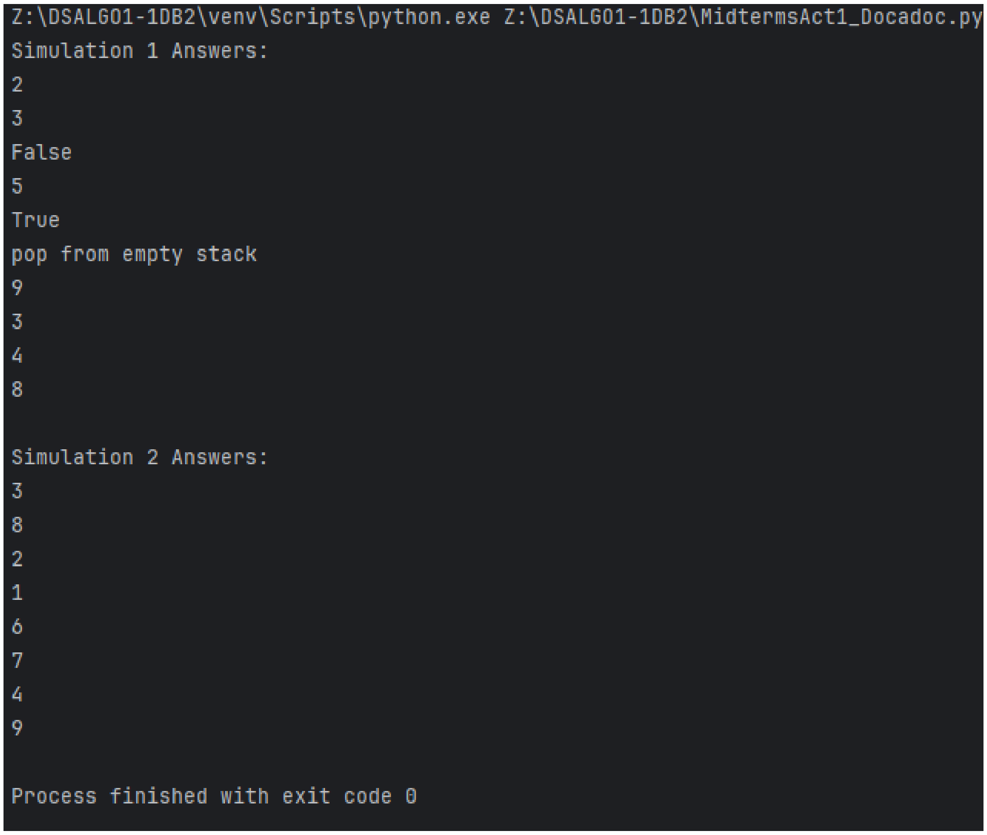
```python
class Stack:
    def push(self, e):
        self.items.append(e)
    def is_empty(self):
        return len(self.items) == 0
    def top(self):
        if self.is_empty():
            raise IndexError("top from empty stack")
        return self.items[-1]
    def pop(self):
        if self.is_empty():
            raise IndexError("pop from empty stack")
        return self.items.pop()
    def __init__(self):
        self.items = []
    def __len__(self):
        return len(self.items)
```



```python
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)
```



```python
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)
```



```
Z:\DSALGO1-1DB2\venv\Scripts\python.exe Z:\DSALGO1-1DB2\MidtermsAct1_Docadoc.py
Simulation 1 Answers:
2
3
False
5
True
pop from empty stack
9
3
4
8

Simulation 2 Answers:
3
8
2
1
6
7
4
9

Process finished with exit code 0
```

```
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)
```