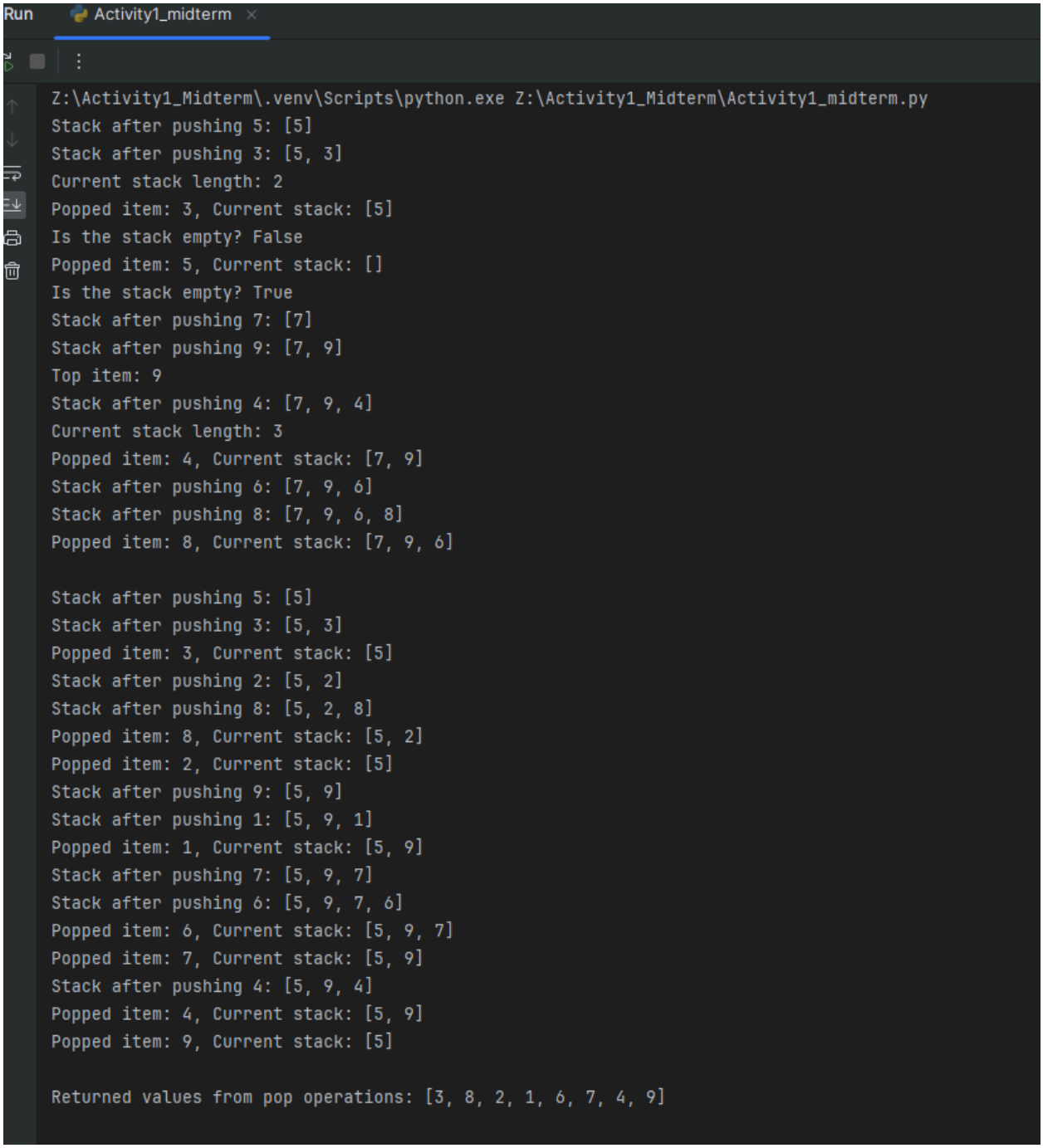
Activity1_midterm.py

```python
2 usages
class Stack:
    def __init__(self):
        self.items = []

    2 usages
    def push(self, item):
        self.items.append(item)
        print(f"Stack after pushing {item}: {self.items}")

    2 usages
    def pop(self):
        if self.is_empty():
            return "Error: Stack is empty"
        item = self.items.pop()
        print(f"Popped item: {item}, Current stack: {self.items}")
        return item

    1 usage
    def top(self):
        if self.is_empty():
            return "Error: Stack is empty"
        return self.items[-1]

    3 usages
    def is_empty(self):
        return len(self.items) == 0

    def __len__(self):
        return len(self.items)


S = Stack()
operations = [
    ("S.push(5)",),
    ("S.push(3)",),
    ("len(S)",),
    ("S.pop()",),
    ("S.is_empty()",),
    ("S.pop()",),
```

```
S = Stack()
operations = [
    ("S.push(5)",),
    ("S.push(3)",),
    ("len(S)",),
    ("S.pop()",),
    ("S.is_empty()",),
    ("S.pop()",),
    ("S.is_empty()",),
    ("S.pop()",),
    ("S.push(7)",),
    ("S.push(9)",),
    ("S.top()",),
    ("S.push(4)",),
    ("len(S)",),
    ("S.pop()",),
    ("S.push(6)",),
    ("S.push(8)",),
    ("S.pop()",)
]


for operation in operations:
    command = operation[0]
    if command.startswith("S.push"):
        value = int(command[7:-1])
        S.push(value)
    elif command == "len(S)":
        print(f"Current stack length: {len(S)}")
    elif command == "S.pop()":
        S.pop()
    elif command == "S.is_empty()":
        print(f"Is the stack empty? {S.is_empty()}")
    elif command == "S.top()":
        print(f"Top item: {S.top()}")

print()
```

```
stack = Stack()
operations2 = [
    ("push(5)",),
    ("push(3)",),
    ("pop()",),
    ("push(2)",),
    ("push(8)",),
    ("pop()",),
    ("pop()",),
    ("push(9)",),
    ("push(1)",),
    ("pop()",),
    ("push(7)",),
    ("push(6)",),
    ("pop()",),
    ("pop()",),
    ("push(4)",),
    ("pop()",),
    ("pop()",)
]


returned_values = []
for operation in operations2:
    command = operation[0]
    if command.startswith("push"):
        value = int(command[5:-1])
        stack.push(value)
    elif command == "pop()":
        returned_values.append(stack.pop())


print("\nReturned values from pop operations:", returned_values)
```

OUTPUT:


Run
Activity1_midterm
Z:\Activity1_Midterm\.venv\Scripts\python.exe Z:\Activity1_Midterm\Activity1_midterm.py
Stack after pushing 5: [5]
Stack after pushing 3: [5, 3]
Current stack length: 2
Popped item: 3, Current stack: [5]
Is the stack empty? False
Popped item: 5, Current stack: []
Is the stack empty? True
Stack after pushing 7: [7]
Stack after pushing 9: [7, 9]
Top item: 9
Stack after pushing 4: [7, 9, 4]
Current stack length: 3
Popped item: 4, Current stack: [7, 9]
Stack after pushing 6: [7, 9, 6]
Stack after pushing 8: [7, 9, 6, 8]
Popped item: 8, Current stack: [7, 9, 6]
Stack after pushing 5: [5]
Stack after pushing 3: [5, 3]
Popped item: 3, Current stack: [5]
Stack after pushing 2: [5, 2]
Stack after pushing 8: [5, 2, 8]
Popped item: 8, Current stack: [5, 2]
Popped item: 2, Current stack: [5]
Stack after pushing 9: [5, 9]
Stack after pushing 1: [5, 9, 1]
Popped item: 1, Current stack: [5, 9]
Stack after pushing 7: [5, 9, 7]
Stack after pushing 6: [5, 9, 7, 6]
Popped item: 6, Current stack: [5, 9, 7]
Popped item: 7, Current stack: [5, 9]
Stack after pushing 4: [5, 9, 4]
Popped item: 4, Current stack: [5, 9]
Popped item: 9, Current stack: [5]
Returned values from pop operations: [3, 8, 2, 1, 6, 7, 4, 9]