1.
Code:

main.py

```
1 usage
class Stack:
    def __init__(self):
        self.item = []

    4 usages
    def is_empty(self):
        return len(self.item) == 0
    7 usages
    def push(self, item):
        self.item.append(item)
        print(f"Pushed item:", item)

    5 usages
    def pop(self):
        if self.is_empty():
            raise IndexError("Cannot pop from an empty stack")
        return self.item.pop()

    1 usage
    def peek(self):
        if self.is_empty():
            raise IndexError("Stack is empty")
        return self.item[-1]

    2 usages
    def len(self):
        return len(self.item)
```

```
S = Stack()

S.push(5)
S.push(3)
print("The current top of Stack: ", S.len())
print("Popped item: ", S.pop())
print("Is the stack empty? ", S.is_empty())
print("Popped item: ", S.pop())
print("Is the stack empty? ", S.is_empty())
try:
    S.pop()
except IndexError as e:
    print(e)
S.push(7)
S.push(9)
print("Top item: ", S.peek())
S.push(4)
print("The current top of Stack: ", S.len())
S.pop()
S.push(6)
S.push(8)
S.pop()
```

Output:

```
Z:\Midterms\.venv\Scripts\python.exe Z:\Midterms\main.py
Pushed item: 5
Pushed item: 3
The current top of Stack:  2
Popped item:  3
Is the stack empty?  False
Popped item:  5
Is the stack empty?  True
Cannot pop from an empty stack
Pushed item: 7
Pushed item: 9
Top item:  9
Pushed item: 4
The current top of Stack:  3
Pushed item: 6
Pushed item: 8

Process finished with exit code 0
```

2.
Code:

```
class Stack:
    def __init__(self):
        self.item = []

    2 usages
    def is_empty(self):
        return len(self.item) == 0
    9 usages
    def push(self, item):
        self.item.append(item)
        print(f"Pushed item:", item)

    8 usages
    def pop(self):
        if self.is_empty():
            raise IndexError("Cannot pop from an empty stack")
        return self.item.pop()

    def peek(self):
        if self.is_empty():
            raise IndexError("Stack is empty")
        return self.item[-1]

    def len(self):
        return len(self.item)

```

```
73
74  S = Stack()
75
76  S.push(5)
77  S.push(3)
78  print("Popped: ", S.pop())
79  S.push(2)
80  S.push(8)
81  print("Popped: ", S.pop())
82  print("Popped: ", S.pop())
83  S.push(9)
84  S.push(1)
85  print("Popped: ", S.pop())
86  S.push(7)
87  S.push(6)
88  print("Popped: ", S.pop())
89  print("Popped: ", S.pop())
90  S.push(4)
91  print("Popped: ", S.pop())
92  print("Popped: ", S.pop())
```

Output:

```
Z:\Midterms\.venv\Scripts\python.exe Z:\Midterms\main.py
Pushed item: 5
Pushed item: 3
Popped:  3
Pushed item: 2
Pushed item: 8
Popped:  8
Popped:  2
Pushed item: 9
Pushed item: 1
Popped:  1
Pushed item: 7
Pushed item: 6
Popped:  6
Popped:  7
Pushed item: 4
Popped:  4
Popped:  9

Process finished with exit code 0
```