

1.

```
1 usage new *
class Stack:
2     new *
3     def __init__(self):
4         self.items=[]
5
6     4 usages new *
7     def is_empty(self):
8         return len(self.items)==0
9
10    7 usages new *
11    def push(self,item):
12        self.items.append(item)
13
14    5 usages new *
15    def pop(self):
16        if not self.is_empty():
17            return self.items.pop()
18        else:
19            return "Stack is empty"
20
21    1 usage new *
22    def top(self):
23        if not self.is_empty():
24            return self.items[-1]
25        else:
26            return "Stack is empty"
27
28    new *
29    def size(self):
30        return len(self.items)
```

```
S = Stack()
S.push(5)
S.push(3)
print("Length:",len(S.items))
print("Popped Element:",S.pop())
print("Stack empty?",S.is_empty())
print("Popped Element",S.pop())
print("Stack empty?",S.is_empty())
print("Popped Element",S.pop())
S.push(7)
S.push(9)
print("Top Element:",S.top())
S.push(4)
print("Popped Element",S.pop())
print("Length: ",len(S.items))
S.push(6)
S.push(8)
print("Popped Element",S.pop())
```

OUTPUT:

```
Length: 2
Popped Element: 3
Stack empty? False
Popped Element 5
Stack empty? True
Popped Element Stack is empty
Top Element: 9
Popped Element 4
Length: 2
Popped Element 8
```

2.

```
1 usage new *
class Stack:
    new *
    def __init__(self):
        self.items=[]
    4 usages new *
    def is_empty(self):
        return len(self.items)==0
    7 usages new *
    def push(self,item):
        self.items.append(item)
    5 usages new *
    def pop(self):
        if not self.is_empty():
            return self.items.pop()
        else:
            return "Stack is empty"
    1 usage new *
    def top(self):
        if not self.is_empty():
            return self.items[-1]
        else:
            return "Stack is empty"
    new *
    def size(self):
        return len(self.items)
```

```
S=Stack()
print()
print()
S.push(5)
S.push(3)
S.push(2)
S.push(8)
print("Popped Element:", S.pop())
print("Popped Element:", S.pop())
S.push(9)
S.push(1)
print("Popped Element:", S.pop())
S.push(7)
S.push(6)
print("Popped Element:", S.pop())
print("Popped Element:", S.pop())
S.push(4)
print("Popped Element:", S.pop())
print("Popped Element:", S.pop())
```

OUTPUT:

```
Popped Element: 8
Popped Element: 2
Popped Element: 1
Popped Element: 6
Popped Element: 7
Popped Element: 4
Popped Element: 9
```