

Exercise: “A” & “G”

```

class stack: 1 usage
    def __init__(self,maxstack):
        self.maxstack = maxstack
        self.stack = []
        self.tos = -1

    def push(self,item): 6 usages
        if self.tos == self.maxstack-1:
            print("Overflow!")
            return
        self.tos += 1
        self.stack.append(item)

    def pop(self): 3 usages
        if self.tos == -1:
            print("Underflow!")
            return
        item = self.stack[self.tos]
        self.stack.pop(self.tos)
        self.tos -= 1
        return item

    def top(self): 2 usages
        return self.stack[self.tos]

    def isEmpty(self): 3 usages
        if self.tos == -1:
            return True
        return False

# Create a stack of maximum size 5
s = stack(5)

# Check if stack is empty
print("Is stack empty?", s.isEmpty())

# Push elements
s.push(10)
s.push(20)
s.push(30)
print("Stack after pushing 3 elements:", s.stack)
print("Top element:", s.top())

# Push more elements
s.push(40)
s.push(50)
print("Stack after pushing 5 elements:", s.stack)

```

```

# Try pushing when full
s.push(60) # should show "Overflow!"

# Pop elements
print("Popped item:", s.pop())
print("Popped item:", s.pop())
print("Stack after popping 2 elements:", s.stack)

# Check current top
print("Current top element:", s.top())

# Empty the stack completely
while not s.isEmpty():
    print("Popped:", s.pop())

# Final check
print("Is stack empty now?", s.isEmpty())

```

“Output”

```

"F:\project 2.0\.venv\Scripts\python.exe" D:\PythonPr
Is stack empty? True
Stack after pushing 3 elements: [10, 20, 30]
Top element: 30
Stack after pushing 5 elements: [10, 20, 30, 40, 50]
Overflow!
Popped item: 50
Popped item: 40
Stack after popping 2 elements: [10, 20, 30]
Current top element: 30
Popped: 30
Popped: 20
Popped: 10
Is stack empty now? True

Process finished with exit code 0

```

```

"F:\project 2.0\.venv\Scripts\pyth
0
5
6
5
X = 5
Y = 4
Z = 5

Process finished with exit code 0

```