Review: STACKS

First Name Last Name

First Name Last Name

First Name Last Name

1. Stack Operations:
What would be the contents of **s1, s2,** and **s3** after the following code is executed?

```
s1.push(10)
s1.push(70)
s1.push(20)
s1.push(30)
s1.push(80)
loop( s1.pop(item) )
    if(item > 50)
        s2.push(item)
    else
        c = s1.getCount()
        s3.push(c)
    end if
end loop
```

2. Stack Operations: Given the following data to be inserted into a stack and the sequence of operations, what is the order that they are processed (popped out of the stack)?

```
Data to be processed through a stack (arrival order):
              10, 20, 30, 40, 50

Operations: push, push, pop, push, push, pop, pop,
            push, pop, pop

Process Order:
```

3. Stack Operations: Given the following set of data to be inserted into a stack and the order that they are processed (popped out of the stack), what sequence of operations must have occurred?

```
Arrival Order: 10, 20, 30, 40, 50
Process Order: 30, 20, 10, 50, 40
Operations:
```

Review: STACKS

4. Stack Applications:

(A). Convert the following expression into its postfix form.

```
50 – 2*(3 + 16/8) – 4*5
```

(B). Convert the following expression into its postfix form.
Use a stack and write the sequence of stack operations.

```
50 – 2*(3 + 16/8) – 4*5
```

Example:

Infix: **2 * 3 + 4 * 5**

Postfix: **2 3 * 4 5 * +**

Operations	Stack at each step (bottom to top)
push	*
pop	//*empty*
push	+
push	+ *
pop	+
pop	//*empty*

5. Stack Applications: Evaluate the following expression in its postfix form.

```
25 2 * 3 12 4 / 9 7 % + * - 30 5 / -
```

Example:

Postfix: **2 3 * 4 +**

Operations	Stack at each step (bottom to top)
push	2
push	2 3
pop, pop	
push:2*3	6
push	6 4
pop, pop	
push:6+4	10

VALUE: **10**