Sheet 3 - Data Structures

Taha Ahmed 19015866

December 2022

1 Question 1

Suppose an initially empty stack S has performed a total of 25 push operations, 12 top operations and 10 pop operations. Three of the pop operations generated "Empty Stack Exception", which were caught and ignored.

1. What is the size of S after performing the operations described above?

25 - 7 = 18

2. If this stack is implemented as an array, what is the value of the "top" data member of the Stack class?

index = length - 1 = 17

2 Question 2

Describe the output of the following series of stack operations: `push(5), push(3), pop(), push(2), push(8), pop(), push(9), push(1), pop(), push(7), push(6), pop(), pop(), push(4), pop(), pop()`.

```
$ push(5)
stack [5]
$ push(3)
stack [5 | 3]
$ pop()
output : 3
stack [5]
$ push(2)
stack [5 | 2]
$ push(8)
stack [5 | 2 | 8]
$ pop()
output : 8
stack [5 | 2]
```

```
$ push(9)
stack [5 | 2 | 9]
$ push(1)
stack [5 | 2 | 9 | 1]
$ pop()
output : 1
stack [5 | 2 | 9]
$ push(7)
stack [5 | 2 | 9 | 7]
$ push(6)
stack [5 | 2 | 9 | 7 | 6]
$ pop()
output : 6
stack [5 | 2 | 9 | 7]
$ pop()
output : 7
stack [5 | 2 | 9]
$ push(4)
stack [5 | 2 | 9 | 4]
$ pop()
output : 4
stack [5 | 2 | 9]
$ pop()
output : 9
stack [5 | 2]
```

3 Question 3

Write an algorithm that returns the number of elements in a stack leaving it unchanged. (Assume that the stack Abstract Data Type provides only pop and push operations).

```
isEmpty(Stack stack):
    try :
        stack.push(stack.pop())
    catch (exeption):
        return true
    return false

getNumberOfElements(Stack stack):
    stack tempStack = new Stack()
    numberOfElements = 0
    while (!isEmpty(stack)):
```

```
        tempStack.push(stack.pop())
        numberOfElements++
    while (!isEmpty(tempStack)):
        stack.push(tempStack.pop())
    return numberOfElements
```

4 Question 4

Write an algorithm that uses a stack to determine if an HTML document is well-formed. (A well-formed HTML document should have all tags properly nested and all opened tags should have the corresponding closing tags).

```
isValidHTML(string HTML):
    Stack stack = new Stack()
    HTML = HTML.split(" ")
    for word in HTML:
        if (isHTMLOpen(word)):
            stack.push(word)
        if (isHTMLClose(word)):
            if (stack.isEmpty()):
                return false
            else if stack.top() != word :
                return false
            else :
                stack.pop()
    return true
```

5 Question 5

A palindrome is a word or a phrase that is the same when spelled from the front or the back. For example reviver and able was I ere I saw elba are both palindromes. Write an algorithm that uses a stack to determine if a word or a phrase is a palindrome.

```
isPalendrom(string phrase):
    Stack stack = new Stack()
    for i = 0 to floor(length(phrase) / 2) :
        stack.push(phrase.charAt(i))
    for i = ceil(length(phrase) / 2) to length(phrase) :
        if stack.pop() != phrase.charAt(i):
            return false
    return true
```

6 Question 6

Write an algorithm that doing the following on the stack leaving it unchanged:

1. Return an identical copy of the Stack.

```
copy(Stack stack)
    Stack tempStack = new Stack()
    Stack copiedStack = new Stack()

    while (!stack.isEmpty()):
        tempStack.push(stack.pop())

    while (!tempStack.isEmpty()):
        element = tempStack.pop()
        Stack.push(element)
        copiedStack.push(element)

    return copiedStack
```

2. Return a reversed copy of the Stack.

```
copy(Stack stack)
    Stack tempStack = new Stack()
    Stack copiedStack = new Stack()

    while (!stack.isEmpty()):
        element = stack.pop()
        tempStack.push(element)
        copiedStack.push(element)

    while (!tempStack.isEmpty()):
        stack.push(tempStack.pop())

    return copiedStack
```

3. Return a sorted copy of the Stack in descending order.

```
sortedCopy(Stack s)
    Stack stack = copy(s)
    Stack tempStack = new Stack()

    while (!stack.isEmpty())
```

```
    temp = stack.pop()

    while (!tempStack.isEmpty() and tempStack.peek() > temp):
        stack.push(tempStack.pop())

    tempStack.push(temp)

return tempStack
```