Course: Data Structures and Algorithms Assignment 2

Stack *Dr. Belal Al-Fuhaidi*

1- Trace the following code, showing the contents of the stack after each invocation:

```
Stack stack = new Stack();
stack.push(new Character('A'));
stack.push(new Character('B'));
stack.push(new Character('C'));
stack.pop();
stack.pop();
stack.push(new Character('D'));
stack.push(new Character('E'));
stack.push(new Character('F'));
stack.pop();
stack.push(new Character('G'));
stack.pop();
stack.pop();
stack.pop();
```

2- Suppose an initially empty **ArrayStack *S*** has performed a total of **25 push** operations, **12 top** operations, and **10 pop** operations, 3 of which returned null to indicate an empty stack. What is the current size of ***S***? And what is the value of the instance variable **t**?

3- Evaluate the following postfix expressions (true or false):

a. 8 2 + 3 * 16 4 / - =
b. 12 2 5 5 1 / / * 8 7 + - =
c. 70 14 4 5 15 3 / * - / 6 + =
d. 3 5 6 * + 13 - 18 2 / + =

4- Convert the following infix expressions to postfix notations, and convert the first two postfix notations to java code using stack operations:

a. (A + B) * (C + D) - E
b. A - (B + C) * D + E / F
c. ((A + B) / (C - D) + E) * F - G
d. A + B * (C + D) - E / F * G + H

5- Write the definition of the function template **printListReverse** that uses a stack to print a linked list in reverse order. Assume that this function is a member of the class **linkedStack**,

6- Write this client method using only the push(), top(), pop(), and isEmpty() methods:

public static <E> void reverse(ArrayStack<E> stack)
// reverses the contents of the specified stack

7- Write this client method using only the push(), top(), pop(), and isEmpty() methods:
public static <E> E popBottom(LinkedStack<E> stack)
// removes and returns the bottom element of the specified stack

8- Add this member method to the ArrayStack class :
public E topSecond()
// returns the second from the top element of this stack

9- Add this member method to the ArrayStack class :
public E popSecond()
// removes and returns the second element of this stack

10- Add this member method to the LinkedStack class:
public E bottom()
// returns the bottom element of this stack

11- Add this member method to the ArrayStack class:
public E popbottom()
// removes and returns the bottom element of this stack

12- **Consider the following segment code with the following informations:**

```
Public static void main (string []args)
     {
      Stack<int>  stack =new ArrayStack (10);

     for (int i=1; i<=10; i++)
                   if (i % 3 != 0)
                    {     stack.push(i* 2);      }
                 else
                      {
                         stack.pop();         }
   }
```

- Assume (capacity = 10, size = 0, top = 0)
  After execution of this code..

a) What are the contents (elements) of the stack?

b) What are the values of the variables <u>count</u>, <u>top</u>?

c) What are the element of the **top( )** method in the stack?

d) Is the stack full? Why?

e) Make the stack return to the empty state?

Good Luck