# 2-Stacks and Queues

**Writing exercises**  
1.   Assume that x, y, z are integer variables and that s is a stack of integers, state the output of each program fragment.  
x = 3;y = 5;z = 2;  
s.makeEmpty( );  
s.push(x);s.push(4);s.pop();  
s.push(y);s.push(3); s.push(z);  
s.pop();s.push(2);s.push(x);  
while(! s.isEmpty( )) System.out.println(s.pop() + "  ");  
2.   x = 3; y = 1;  
s.makeEmpty();  
s.push(5);s.push(7);s.pop();  
x += y;  
s.pop();  
s.push(x);s.push(y); s.push(2);s.pop();s.pop();  
while (! s.isEmpty( ))  
{  
   y = s.pop();  
   System.out.println(y);  
}  
System.out.println("x = " + x);  
System.out.println("y = " + y);

3.   A letter means push and an asterisk means pop in the  following sequence.  Give the sequence of values returned by the pop operations  when this sequence of operations is performed on an initially empty stack.

         E A S \* Y \* Q U E \* \* \* S T \* \* \* I O \* N \* \* \*

4.   A letter means push and an asterisk means pop in the following sequence.  Give the contents of s[0], ..., s[4] after  this sequence of operations is performed on an initially empty stack (the stack is implementated by array s[]).

         L A \* S T I \* N \* F I R \* S T \* \* O U \* T \* \* \* \* \* \*

5**.   A letter means enqueue and an asterisk means dequeue in the  following sequence.  Give the sequence of values returned by the dequeue operation when this sequence of operations is performed on an initially empty queue.**

**E A S \* Y \* Q U E \* \* \* S T \* \* \* I O \* N \* \* \***

**6.   A letter means enqueue and an asterisk means dequeue in the following sequence.  Give the contents of q[0], ..., q[4] after  this sequence of operations is performed on an initially empty queue (the queue is implementated by circular array q[] with size 5).**

**E A S \* Y \* Q U E \* \* \* S T \* \* \* I O \* N \* \* \***

**Practical exercises**  
**Question 1.** Write a Java program to implement a **stack** of integer values with the following operations:  
1.   boolean isEmpty() - return true if the stack is empty and false otherwise.  
2.   **void clear() - clear the stack.**

3.   void push(int x) - insert  a node with value  x  at the top of  the stack.  
4.   int pop() - remove the top element on the stack and return it's value; throw EmptyStackException for empty stack.  
**5.   int top() -  return value of a node at the top of the stack; throw EmptyStackException for empty stack.**  
**6.**   **void traverse() - display all values in the stack from the top to the bottom.**7.   Use a stack to convert an integer number in decimal system to binary system and display on the screen.  
Note: You can write some constructors and other methods as you see they are necessary.

**Question 2.** Write a Java program to implement a **queue** of integer values with the following operations:  
1.   boolean isEmpty() - return true if the queue is empty and false otherwise.  
2.   void clear() - clear the queue.  
3.   void enqueue(int x) - insert  a node with value  x  at the end of  the queue.  
4.   int dequeue() - remove the first element on the queue and return it's value; throw Exception for empty queue.  
5.   int first() -  return value of  the first node of the queue; throw Exception for empty queue.  
4.   void traverse() - display all values in the queue from the front to the rear.  
6.   **Use a queue to convert a real number less than 1  in decimal system to binary system and display on the screen.**Note: You can write some constructors and other methods as you see they are necessary.

**~~Question 3.~~**~~Modify rograms in questions 1 and 2 so that elements in the stack and queue are String types  and write the Main class to test them.~~

**~~Question 4.~~**~~Modify rograms in questions 1 and 2 so that elements in the stack and queue are character  types  and write the Main class to test them.~~

**Question 5.** Modify rograms in questions 1 and 2 so that elements in the stack and queue are Object types and write the Main class to test them (e.g. stack and queue contains computer names).