Question 3 ..... 20 points

Write the method **public** <T> **boolean** isReverse(Queue<T> q1, Queue<T> q2), user of the ADT Queue. The method checks if q2 is the reverse of q1. The two queues must not change after the call.

1. Line 1:
   - (A) **if** (q1.length()== q2.length())
   - (B) **if** ((q1.length()== 0)&& (q2.length()==0))**return true**;
   - (C) **if** (q1.length()!= q2.length())**return false**;
   - (D) **if** ((q1.length()== 0)|| (q2.length()== 0))**return true**;
   - (E) None
2. Line 2:
   - (A) Stack<T> x1 = **new** Stack<T>(), x2 = **new** Stack<T>();
   - (B) Queue<T> x1 = **new** LinkedQueue<T>(), x2 = **new** LinkedQueue<T>();
   - (C) Stack<T> x1 = **new** LinkedStack<T>(), x2 = **new** LinkedStack<T>();
   - (D) Stack<T> x = **new** LinkedStack<T>();
   - (E) None
3. Line 3:
   - (A) **while** (!q1.empty())
   - (B) **while** (q1.length()!= 0)
   - (C) **while** (x1.length()!= 0)
   - (D) **while** (!x.full ())
   - (E) None
4. Line 4:
   - (A) x.push(q1.retrieve());
   - (B) x1.push(current.data);
   - (C) x1.push(q1.serve());
   - (D) x1.enquque(q1.serve());
   - (E) None
5. Line 5:
   - (A) **while** (!x.empty()){
   - (B) **while** (x1.size()>0){
   - (C) **boolean** eq = **false**;
   - (D) **boolean** eq = **true**;
   - (E) None
6. Line 6:
   - (A) T e1 = x1.pop(), e2 = x2.pop();

(B) `T e1 = x.pop(), e2 = q2.serve();`

(C) **while** `(!x1.empty()){`

(D) **while** `(x1.size()>0){`

(E) None

7. Line 7:

(A) **if** `(!e1.equals(e2))`**break**`;`

(B) `T e1 = x1.pop(), e2 = q1.serve();`

(C) **if** `(!e1.equals(e2))`

(D) `T e1 = x1.pop(), e2 = q2.serve();`

(E) None

8. Line 8:

(A) **if** `(!e1.equals(e2))`**continue**`;`

(B) **if** `(!e1.equals(e2))eq = false;`

(C) `x2.push(e1); q2.enqueue(e1); }`

(D) **return false**`;`

(E) None

9. Line 9:

(A) `x.push(e1); q2.enqueue(e2); }`

(B) `x2.push(e1); q2.enqueue(e2); }`

(C) `x1.push(e1); q2.enqueue(e1); }`

(D) **return false**

(E) None

10. Line 10:

(A) **while** `(!q2.empty())`

(B) **while** `(!x2.empty())`

(C) **while** `(!q1.full())`

(D) **while** `(!x.last())`

(E) None

11. Line 11:

(A) `q1.enqueue(x1.pop());`

(B) `q1.enqueue(x2.pop());`

(C) `q2.enqueue(x.pop());`

(D) `q2.enqueue(x2.pop());`

(E) None

12. Line 12:

Ⓐ `return false;`

Ⓑ `return true;`

Ⓒ `return e1.equals(e2);`

Ⓓ `return eq;`

Ⓔ None