Tutorial 8 - Queues

1. Please refer to the question below:
(a) How would you implement a stack using two STL queues that is efficient in pop() operation?
(b) How would you implement a queue using two STL stacks?

2. Let SQ be a class supporting the following operations:

```
void push (const int& t);
void pop(int& t);
```

Describe a way to determine whether SQ behaves like a stack or a queue.

3. Write the code of the template functions push_front and pop_back for the circular queue.

```
template<typename  T>
void Queue<T>::push_front  (canst T& newItem) {        }

template  <typename T>
void Queue<T>:: pop_back ( T& queueBack){   }
```

4. Big decimal can be represented in a queue with each segment of 4 decimals stored in one element of the queue having the least significant segment as the front element. For example, 1234567890 can be stored in a 3-element queue as 12, 3456, 7890 with 7890 being the front of the queue, followed by 3456, and 12 is the last element. Let us call such a queue a big decimal queue (BDQ). Implement the function to multiply a BDQ with an integer that is less than 10000. The function should return the product in a BDQ. Implement a test driver as well.

5. Write a function that returns the second largest integer in an STL queue of integer without destroying the queue. For example, in the queue {1, 11, 2, 3, 10, 4}, the function will return 10. Assuming there are always 2 or more elements in a queue. Implement a test driver.

6. Implement the following function with the use of a queue of characters to store the input line as it is being read, one character at a time.

```
unsigned int counter( )
  // Precondition: None
  // Postcondition: A line of input has been read from cin, up to but
  // excluding the newline character. White spaces are ignored.
  // The return value of the function is the number of times that the Last
  // character of the line appeared somewhere in the line.
  // For example, when the input is “ABBX        DXDZX”, then
  // the value returned is 3 since there are 3 X's in the input line.
```