CSC 212 Tutorial
Queues & Stacks

Problem 1

Write the recursive static method *split* that splits a queue of n elements into two queues. The elements with odd orders (i.e. $1st, 3rd, 5th$ ...) should be put in the first queue and elements with even orders (i.e. $2nd, 4th, 6th$ ...) should be put in the second queue. The original queue should remain unchanged at the end of the method. The method signature is: **public static** `<T>` **void** `split(Queue<T> q, Queue<T> oq, Queue<T> eq)`.

Example 1.1. *Given the queue* (A, B, C, D, E)*,* split *results in* oq (A, C, E)*, and* eq (B, D)*.*

Problem 2

Write a static method *remove* that removes every element in the priority queue having priority less than p. The method signature is **public static** `<T>` **void** `remove (PriorityQueue<T> pq,` **int** `p)`.

Example 2.1. *Given* pq*:* $[A, 10], [D, 8], [B, 5], [E, 3], [C, 2]$ remove(pq, 5) *results in* pq*:* $[A, 10], [D, 8], [B, 5]$.

Problem 3

Write the recursive static method *search* that searches for an element e in a stack st and returns true if it's found or false otherwise. st should not change at the end of the method. You are not allowed to use any auxiliary data structures. The method signature is: **public static** `<T>` **boolean** `search(Stack<T> st, T e)`

Example 3.1. *Given the stack* st *(top-to-bottom):* $5, 7, 5, 3, 2$. search(st, 3) *returns true while* search(st, 1) *returns false.*