CSC 212 Tutorial #6
Stack

Problem 1

Write the static method *replace*, user of *Stack* ADT, that takes as input a stack *st* and two elements *e1* and *e2* and replaces all the occurrences of the element *e1* in *st* with *e2*. The method signature is: `public static<T> void replace(Stack<T> st, T e1, T e2)`

Example 1.1. *Given the stack* st *(top-to-bottom):* $5, 7, 5, 3, 2$. replace(st, 5, 0) *results in:* $0, 7, 0, 3, 2$.

Problem 2

Write the recursive static method *copy* that takes two stacks *st1* and *st2* and copies all the elements of *st1* into *st2* in the same order while preserving *st1*. You can assume *st2* can hold all *st1* elements. You are not allowed to use any auxiliary data structures. The method signature is: `public static <T> void copy(Stack<T> st1, Stack<T> st2)`

Example 2.1. *Given the stack* st1 *(top-to-bottom):* $4, 7, 1, 5, 2$ *and* st2*:* $\varnothing$. copy(st1, st2) *results in* st2*:* $4, 7, 1, 5, 2$.

Problem 3

Write the recursive static method *search* that searches for an element *target* 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 target)`

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.*