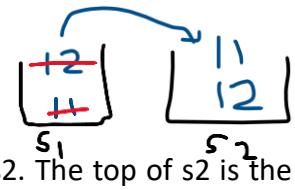
Lab 10

Note: You can reuse your basic stack program for all these programs. For this

#include "filename.c"

1. Write functions to implement a queue like behaviour using two stacks. This means a FIFO behaviour is to be implemented on LIFO data structures.
    a. Use stack, s1, to enqueue the elements. When you need to enqueue, push onto s1. O(1) complexity

    b. **To simulate "first out"**-> Use stack, s2, to dequeue.



        i. <u>If s2 is not empty</u>, pop from s2.
        ii. <u>If s2 is empty</u> then move elements from s1 to s2. The top of s2 is the "first in" element. POP from s2 so FIFO is successful!!!
        iii. What is complexity of your code?
2. Write functions to implement a queue like behaviour using a single stack. **You will have to use recursion for this.** Recursion will reorder the elements during the dequeue operation. Perform the following:
    a. For enqueue: push onto stack. O(1) complexity
    b. For dequeue: create a recursive function that returns an integer. Do the following in sequence:
        i. pop the top of the stack and store in a local element, *ele*
        ii. Write the base condition with of an empty stack and return *ele*.
        iii. Make the recursive call. Remember, your call returns an integer and so store that in another local variable, *temp*.
        iv. Push onto stack, ele.
        v. Return *temp*

    How does this program work? Understand it by writing the pseudocode and drawing the stack frame of each recursive call on paper first. After you have understood it, code the steps mentioned. <u>Run for code by enqueueing 12,22,23,24,25 and one dequeue operation.</u>

    c. What is the complexity of your dequeue operation?