New Beginnings – Summer 2018

C++ Programming - Queue/Stack Practice

Create a new Palindrome checker using two data structures

1. One queue
2. One stack

The code should be broken into two functions with:

1. bool isPalindrome(char *);
    a. Takes a pointer to a character array
    b. Returns a bool.
        i. True is the character array is a palindrome
        ii. False if not
    c. Use a Stack and a Queue to implement the checker
        i. The Stack and Queue should be in their own classes
            1. stack.cpp/stack.hpp
                a. Methods:
                    i. void push(char);
                    ii. char pop();
            2. queue.cpp/queue.hpp
                a. Methods:
                    i. void enqueue(char);
                    ii. char dequeue();
    d. The dequeue and pop functions should free the node memory as we saw in class today. To check for memory leaks, use valgrind by running
        i. %valgrind –leak-check=full <program name>

2. int main();
    a. Asks the user for a file name.
    b. Opens the file and reads it line by line. Each line contains:
        i. <word> <0/1 flag>
        ii. 0 if the word is NOT a palindrome
        iii. 1 if the work IS a palindrome
    c. Call the isPalindrome() function with the word as the parameter and compare the output of the function to the 0/1 flag from the file.
    d. For each word, output the word, if it is a palindrome(per your function) and if it matches the 0/1 flag.