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 \*);
   1. Takes a pointer to a character array
   2. Returns a bool.
      1. True is the character array is a palindrome
      2. False if not
   3. Use a Stack and a Queue to implement the checker
      1. The Stack and Queue should be in their own classes
         1. stack.cpp/stack.hpp
            1. Methods:

void push(char);

char pop();

* + - 1. queue.cpp/queue.hpp
         1. Methods:

void enqueue(char);

char dequeue();

* 1. 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
     1. %valgrind --leak-check=full <program name>

1. int main();
   1. Asks the user for a file name.
   2. Opens the file and reads it line by line. Each line contains:
      1. <word> <0/1 flag>
      2. 0 if the word is NOT a palindrome
      3. 1 if the work IS a palindrome
   3. 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.
   4. For each word, output the word, if it is a palindrome(per your function) and if it matches the 0/1 flag.