

Use of Stacks

In this programming assignment you will use stacks to evaluate if a given string is a palindrome. A palindrome is a string that reads the same forward and backwards. Examples include “Madam, I’m Adam” or “tacocat”. Punctuation is ignored in determining if an expression is a palindrome.

Your code will **not** be recursive. **In your analysis, suggest a recursive algorithm and compare it with your stack-based solution.** The purpose of using a stack is to take advantage of its LIFO nature, therefore algorithms which merely use the stack for storage and determine inclusion of the string in the language by the use of counting the input string in any manner will **NOT receive any credit.** Your algorithm should be based on stack manipulation, and you may use stacks as extensively as you see fit. In processing a string to see if it is a palindrome you may use one stack or several. You may not use any other data structure.

Test each string given **as well as additional strings you make up yourself.**

Plan to read each line character by character, ignoring punctuation and pushing the valid characters onto a stack. Evaluate the expression and return the result before reading the next line.

This is a straightforward assignment so to make it more interesting you need to implement the stack using a queue as underlying data structure.

Plan your code on paper completely and debug it on paper completely before starting development. It may make sense to implement the palindrome validation using library stacks and once that is working then implement your own stacks based on queues. You may use an array-based queue or a linked queue as you prefer. Start with a stack ADT to make sure you cover all the necessary stack features with your queue based implementation of a stack.

Be sure to discuss your data structures and their implementation and why they make sense. E.g. why is stack a reasonable choice to solve this problem? What implementation of a queue did you choose? Why? As stated above, consider a recursive solution and compare it to your iterative solution. Is one better than the other? If so, why? Your review of a recursive solution should include basic approach/algorithm and comparison to your implemented solution, as well as motivation. You do not need to create a detailed alternative implementation.

Be sure to review the Programming Assignment Guidelines for specific requirements for the Analysis and before submitting this assignment.

Note: **You are expected to write the stack/queue code yourself** and not use the library queue class. Be sure to **include the code for your stack** as part of the source code you submit.

Your code should always allow for the specification of input and output file names, so that it can be tested against different input sets. Do not hardcode the file names into your code. It is acceptable to prompt the user for the file names, if you have to. It is preferred to pass in the file names as command line parameters.

In the required input file

“Able was I ere I saw Elba” (1848) Alludes to the first exile of Napoleon to Elba

“A man, a plan, a canal – Panama” (1948) By Leigh Mercer, published in *Notes and Queries*, 13 November 1948, according to *The Yale Book of Quotations*, F. R. Shapiro, ed. (2006, [ISBN 0-300-10798-6](#)).

“Sator Arepo Tenet Opera Rotas” is a Latin graffito found at [Herculaneum](#), a city buried by ash in 79 CE. This translates as “The sower Arepo holds with effort the wheels”.

“Doc, note: I dissent. A fast never prevents a fatness. I diet on cod” ["Professor Peter Hilton"](#). [Daily Telegraph](#). London. 10 November 2010.

“T. Eliot, top bard, notes putrid tang emanating, is sad; I'd assign it a name: gnat dirt upset on drab pot toilet” By Brendan Gill, published in *Here At The New Yorker*, (1997, [ISBN 0-306-80810-2](#)).