

# CS270 Programming Assignment 6

## “Recursive Palindrome Checking”

Program due Friday, April 19 via RamCT by 1 PM.

### Goals

Write a C program that checks a string to see if it is a palindrome. It must use a recursive function for checking that uses pointers to access and compare two characters.

You will learn how to implement a solution recursively and will learn how to use pointers in C.

### Program Structure Specification

Your program should use a `main()` function with a recursive function `isPalin()`. `Palin` will call itself until it is determined that the string is a palindrome, or that it is not. It accepts an input string of up to 40 ASCII characters (ensure somehow that it will handle spaces), and places it in an array starting at location `ZStr`. Use a function `areSame()` to determine if two characters are considered to be the same. Use a function `isSkippable()` to determine if a character is a non-letter.

You must use pointers for accessing the elements of the array (not array index). You may not use any of the functions in `string.h`. Uppercase and lowercase letters are considered equivalent, and any spaces or non-letters (such as punctuation mark) in the string should be ignored. Thus "Never odd or even" is a palindrome.

In at least one of the functions, at least one of the arguments must be a pointer.

Include a test case and the response at the end in form of a comment. Also in the comments, acknowledge any web documents you may have consulted. Give the URL and mention what you found there. Finally also include the time you spent in each of the design, coding and testing phases for your program.

## Submission Instructions

- Submit the pa6.c file will be submitted directly to RamCT.

## Reminders:

1. Include appropriate information at the top of the code.
2. Adds comments appropriately.
3. Illegal inputs (when the input string is too long) should generate message "Illegal Input".

## Grading Criteria

To grade the assignment, we will examine and run program (30 points), and we will verify that your computation gets the right answers on the example data and our own test data (35 points). In addition points will be given for coding style and comments (15 points), following assignment directions (20 points). No points if you do not use recursion and pointers. The grading factors we consider for coding style include having clear and concise comments, consistent indentation, and the minimal amount of code to solve the problem.

## Late Policy

There is a 25% penalty for late submission. They cannot be more than 24 hours late.