

## Problem Set 4 Exercise #16: Palindromes

**Reference:** Lecture 11 notes

**Learning objectives:** Characters and Strings; Algorithm design

**Estimated completion time:** 50 minutes

### Problem statement:

A **palindrome** is a word or phrase that can be read the same way in either direction, ignoring case, punctuation and spaces. The following are just a few examples that start with 'A', taken from [www.palindromelist.net](http://www.palindromelist.net) where you can find a huge collection:

- A car, a man, a maraca.
- A nut for a jar of tuna.
- Are we not drawn onward to new era?

Write a **non-OO** program **PS4\_Ex16\_Palindromes.java** to read in a list of strings, and for each input string, determine whether it is a palindrome or not. Your program output is the number of palindromes found.

Your program should include a static method

```
boolean isPalindrome(String str)
```

that returns true if **str** is a palindrome, or false otherwise. When determining whether **str** is a palindrome, only alphabet ('a' to 'z', and 'A' to 'Z') count ignoring case. All other characters such as punctuation marks, digits and spaces should be ignored.

### Sample run #1:

```
How many strings? 5
Enter 5 strings, each on a line:
A car, a man, a maraca.
Yeh boil! I obey!
Name no one man.
Ya, Decaf. FACE DAY!!
Map, DNA, and spam.
Number of palindromes = 3
```

### Sample run #2:

```
How many strings? 7
Enter 7 strings, each on a line:
One Two Three Four Five
Ten animals I slam in a net.
```

```
Decaf and DNA faced.  
Go dog.  
Elite tile.  
Today.  
Borrow or rob?  
Number of palindromes = 5
```

**Useful tips:**

1. Explore the `isLetter()` and `toLowerCase()` methods of the Java `Character` class and the `length()` and `charAt()` methods of the `String` class.
2. Pay attention to the whitespace issue as encountered in Ex #12.
3. Copy the input from the PDF file and paste them into the interaction panel of DrJava when running your program.