CS 321 Homework 0 – Attempt by Tuesday, Jan. 17, but do NOT hand in!

Checking for Palindromes

A sequence of characters is a palindrome if it reads the same backwards as forwards. For example, "`eye`", "`racecar`", "`abba`", and "`xxyyxx`" are all palindromes. Every single character is a palindrome, and so is the empty sequence. (Note that there's no need for the sequence to be a "real" word from some dictionary.) We can get more entertaining palindromes if we ignore word breaks, punctuation and capitalization; famous examples include "`Madam, I'm Adam`" and "`A man, a plan, a canal: Panama`".

Your task is to write a Java class, `PalTest`, that tests a character string to see if it is a palindrome. The string to be tested is given in the command line arguments; that is, if your Java class has method

```
public static void main(String argv[]);
```

then the string to be tested is the concatenation of the entries in array `argv`. (Exactly how the command line arguments are set depends on your shell or other execution environment. Usually, each "word" specified on the command line will appear as a separate element of `argv`, where "words" are separated by spaces or tabs. But this can be changed by various quoting mechanisms in the shell. Also, some characters have special meaning in the shell, so they must be "escaped" for them to be seen by the Java program.) Your test for being a palindrome should *ignore* any non-alphabetic characters (including whitespace) and also the capitalization of characters. If the input represents a palindrome, the program should print the string `true` to standard output; otherwise it should print the word `false`. For example, using a unix-like shell:

```
% java PalTest abba
true
% java PalTest abracadabra
false
% java PalTest a man, a plan, a canal: Panama
true
%
```

The purpose of this exercise is to get you going using Java and the specific compiler and JVM on your system. This exercise will *not* be graded and should *not* be submitted.

There are several possible algorithms for detecting palindromes. Once you've chosen an algorithm, there will be many ways to code it in Java. You may well find useful routines in the Java libraries; in particular, take a look at the classes `String`, `StringBuilder`, and `Character` in package `java.lang`.