Problem Set 4 Exercise #09: Palindromes

Reference: Lecture 10 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 and any non-letter characters. The following are just a few examples that start with 'A', taken from 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 program **palindromes.c** to read in a list of strings, each on a line, 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 function

```
int is_palindrome(char str[])
```

that returns 1 if **`str`** is a palindrome, or 0 otherwise.

You may assume that there is at least one letter in every input string, and there is at least one input string. You may also assume that an input string is no longer than 80 characters.

Note that 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.

Some tips are given at the end of next page.

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 **`isalpha()`** and **`tolower()`** functions of **`<ctype.h>`**.
2. Pay attention to the whitespace issue as you have encountered in Ex #03.
3. You may use two indices to point to the *front* and *back* positions of the string. Move the indices inwards when comparing, ignoring non-letters. This programming skill can be used in some other exercises as well.