# ICA09 - More practice with strings

## Product Code Validator

Write a program that asks the user to enter a product code for a product manufactured by the Wacky Wally Widget Company. Your program should inspect the product code provided by the user and respond with the string "Valid" or "Not Valid".

Valid product codes are of the pattern 99.AAAA where "99" is any sequence of two digits and "AAAA" is any sequence of exactly four alphabet characters. The Wacky Wally Widget Company treats upper and lower-case letters in their product codes as the same. That is, "99.AAAA" is valid, and "99.Aaaa" is valid, and "99.aaaa" is also valid, etc.

All of those variations on "99.AAAA” all refer to the same product.

## Getting started (Together as a class)

1. Retrieve ica09\_productcode.zip provided by your instructor.
2. Extract the code into your mis2402workspace and open the wackywally.html file in Visual Studio Code.
3. Write the code in isWWProductCode that will test to see if the parameter text contains a valid product code. The function should return the text “Valid” or “Not Valid” appropriately.

NOTE: There are lots of acceptable ways to solve this problem. But this exercise is meant to give you a chance to put your knowledge various string properties to work. Any approach that requires you to use a string method that has not been covered in class is not advised!

1. Here are some tips to help you work through this problem.

|  |
| --- |
| * 1. It would be smart to convert the whole input string to upper case.
	2. How long is a valid product code? You can check for that. If the value of text is the wrong length, then return “Not Valid”. (Recall that using “return” causes the function to quit immediately!)
	3. What character separates the numeric and alphabetic portions of product code? Check for that. Notice that, given the pattern shown to you, that special character always appears in the same place. If that character does not appear in the expected postion, then return “Not Valid”.
	4. Check each of the first two characters of the text variable. Are they digits? (The parseInt() function can help you do that. If the parseInt(x) is equivalent to x, then x is an integer.
	5. Check the last four characters of the text variable. They should be alpha characters “A” though “Z”

HOT TIP: You can used comparison operators on characters as well as numbers… so the character “B” is less than the character “C” but greater than the character “A” |
|  |  |

1. Test your work. Sample screenshots are shown below.

 

 

 

1. Test your work, again.
2. Upload your wackywally.html file to the corresponding assignment on canvas.

# On your own / Optional

## Palindrome Checker

A palindrome word is a word that is spelled the same backwards and forwards.

Some valid palindrome words: Racecar, madam, Kayak

Notice that in the above examples the case of the letters does not matter ... Racecar is still a palindrome word even though the "R" and "r" are technically different characters.

Write a program that checks to see if a word is a palindrome.

You can be very liberal in your definition of a "word" here... a word is just any pattern of characters. So, abcdef and abccba are both words. The word abcdef is not a palindrome. The word abccba is a palindrome.

More examples:

* “madam” is a palindrome.
* “Nurses run” is not a palindrome according to our definition. (The space throws it off.)

The isPalindrome function you write should return a string. The string should be either "Palindrome" or "Not a palindrome"

1. Retrieve optional\_palindrome.zip provided by your instructor.
2. Extract the code into your mis2402workspace and open the palindromechecker.html file in Visual Studio Code.
3. Complete the isPalindrome function so that it returns either “Palindrome” or “Not a palindrome” appropriately.
4. Test your work.