**Uncovering Hidden Palindromes**

**Exercise:** Uncovering Hidden Palindromes

**Challenge:** Write a pseudocode program in your preferred language to solve the following problem.

Given a string s, **find and count all the palindromic substrings.** A palindrome is a word, phrase, number, or other sequence of characters that reads the same forward and backward, like “madam” or “noon”.

The substrings should be case-insensitive and consist of only alphanumeric characters.

It is not necessary to compile the program. You will be asked to explain your algorithm and provide a sketch to illustrate your solution.

**Requirements:**

* A palindrome can be a single character, such as “a” or “7”.
* The function should return the count of palindromic substrings and a list of all the palindromic substrings.
* Do not count duplicate palindromes (if a palindrome appears multiple times, count it only once).
* Case-insensitive: “a” and “A” are the same palindrome.
* The order of palindromic substrings in the list does not matter.
* The input string may contain spaces and punctuation, but these should be ignored when determining palindromic substrings.

Examples:

| **INPUT** | **OUTPUT** | |
| --- | --- | --- |
| **Count** | **palindromic substrings** |
| ‘abba’ | 4 | [‘a’,’abba’,’b’,’bb’] |
| ‘abc’ | 3 | [‘a’, ‘b’, ‘c’] |
| ‘abcabc’ | 3 | [‘a’, ‘b’, ‘c’] |
| ‘abcABC’ | 3 | [‘a’, ‘b’, ‘c’] |
| ‘madam’ | 5 | [‘a’, ‘ada’, ‘d’, ‘m’, ‘madam’] |
| ‘noon’ | 4 | [‘n’, ‘noon’, ‘o’, ‘oo’] |
| ‘race a car’ | 5 | [‘a’, ‘aca’, ‘c’, ‘e’, ‘r’] |
| ‘amimemima’ | 9 | [‘a’, ‘amimemima’, ‘e’, ‘i’, ‘imemi’, ‘m’, ‘mem’, ‘mim’, ‘mimemim’] |
| ‘a1b2c3’ | 6 | [‘1’, ‘2’, ‘3’, ‘a’. ‘b’, ‘c’] |
| ‘A man, a plan, a canal, Panama!” | 18 | [‘a’, ‘aca’, ‘ama’, ‘amanaplanacanalpanama’, ‘ana’, ‘anacana’, ‘anaplanacanalpana’, ‘aplanacanalpa’, ‘c’, ‘l’, ‘lanacanal’, ‘m’, ‘manaplanacanalpanam’, ‘n’, ‘nacan’, ‘naplanacanalpan’, ‘p’, ‘planacanalp’] |

Good luck!