Author

[madhuram\_j](https://codefights.com/profile/madhuram_j)

https://codefights.com/img/coins_new.png2000

We call letter x a *counterpart* of letter y, if x is theith letter of the English alphabet, and y is the (27 - i)th for each valid i (1-based). For example, 'z' is the *counterpart* of 'a' and vice versa, 'y' is the*counterpart* of 'b', and so on.

A *properly closed bracket word* (*PCBW*) is such a word that its first letter is the *counterpart* of its last letter, its second letter is the *counterpart* of its last by one letter, and so one.

Determine if the given word is a *PCBW* or not.

**Example:**

* ClosedBracketWord("abiryz") = true
  + 'a' is the *counterpart* of 'z';
  + 'b' <-> 'y'
  + 'i' <-> 'r'
* ClosedBracketWord("aibryz") = false
* ClosedBracketWord("abitryz") = false
* **[input] string word**
  + A string consisting of lowercase letters, 0 ≤ len(word) ≤ 20.
* **[output] boolean**
  + true if word is a *PCBW*, false otherwise.

<https://codefights.com/challenge/wTTqjvjHmHZ4z6abQ>

bool ClosedBracketWord(string word)

{

Dictionary<char, char> diccio = new Dictionary<char, char>();

diccio['a'] = 'z';

diccio['b'] = 'y';

diccio['c'] = 'x';

diccio['d'] = 'w';

diccio['e'] = 'v';

diccio['f'] = 'u';

diccio['g'] = 't';

diccio['h'] = 's';

diccio['i'] = 'r';

diccio['j'] = 'q';

diccio['k'] = 'p';

diccio['l'] = 'o';

diccio['m'] = 'n';

diccio['n'] = 'm';

diccio['o'] = 'l';

diccio['p'] = 'k';

diccio['q'] = 'j';

diccio['r'] = 'i';

diccio['s'] = 'h';

diccio['t'] = 'g';

diccio['u'] = 'f';

diccio['v'] = 'e';

diccio['w'] = 'd';

diccio['x'] = 'c';

diccio['y'] = 'b';

diccio['z'] = 'a';

int i = 0, j = word.Length - 1;

while (i <= j)

{

if (word[j] != diccio[word[i]])

{

return false;

}

i++;

j--;

}

return true;

}