SCHOOL
SUPERIOR
MEDIA
ARTS
AND DESIGN
POLYTECHNIC
FROM PORT

ALP

TEACHING ACTIVITY

Algorithms and Data Structures

CURRICULAR UNIT

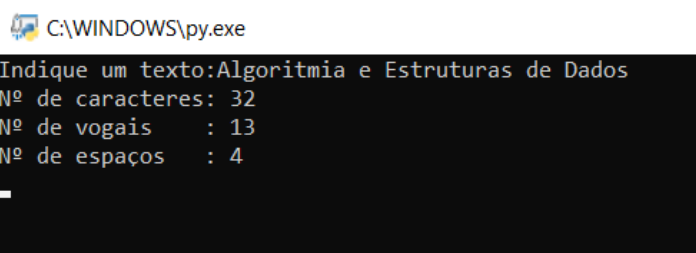| **Sheet 03 – Strings & Functions** |
|---|

1. Implement the **invertText**(*text*) function. The function should receive a text (string), based on an input, and should print the same text but in reverse order.


```
C:\WINDOWS\System32\cmd.    ×    +   ∨

Indique um Texto: Algoritmia e Estruturas de Dados
sodaD ed saruturtsE e aimtiroglA
Press any key to continue . . .
```

2. Implement the **countText**(*text*) function. The function should receive a text, based on an input, and should print:
   - The number of characters
   - The number of spaces
   - The number of vowels

   Included in this text.

C:\WINDOWS\py.exe

```
Indique um texto:Algoritmia e Estruturas de Dados
Nº de caracteres: 32
Nº de vogais    : 13
Nº de espaços   : 4
```

3. Create the function **capicua**(*text*) that **receives** a text as an input parameter and **returns** True or False, depending on whether the text is a capicua or not.
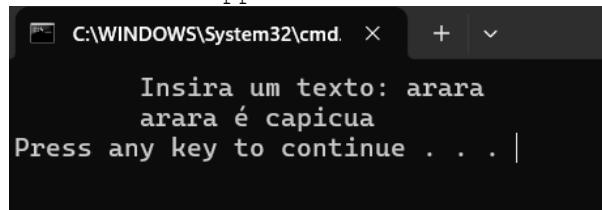
   A *capicua* or *palindrome* word consists on text that can be read from left to right as from right to left.

   Examples of palindromes words: *osso, asa, ana, arara*
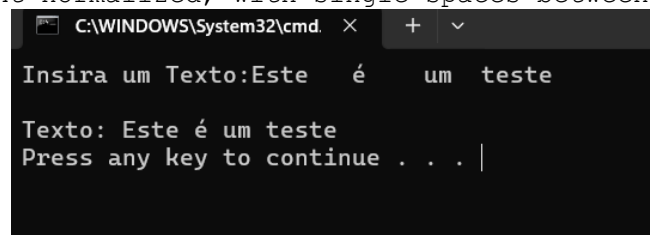
   Examples of the usage of capicua function:
   capicua('osso') => returns True
   capicua('roma') => returns False

Depending on the value returned by the function (True or False), you should print in the console application if the word is capicua or not.

4. Write the **removeSpaces**(*text*) function that **receives** a text and replaces sequences of two or more spaces with a single space. The function should **return** the text normalized, with single spaces between words.
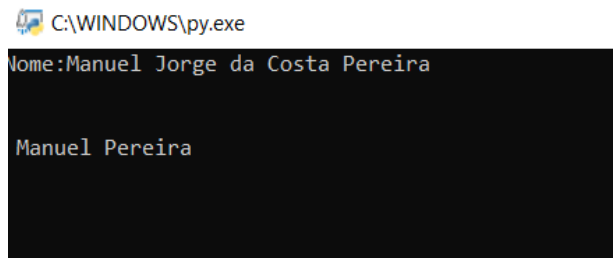

5. Write the function **shortName**(*name*) that should **receive** a full name (based on an input) and **return** a string with the first and last name (first given name and last surname).

   Example:
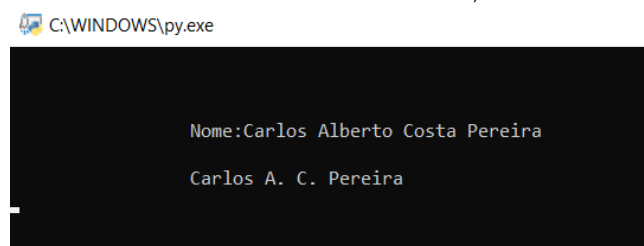   **shortName**('Manuel Jorge da Costa Pereira') => Manuel Pereira

6. Create the function **standardName**(*name*) that should **receive** a full name (based on as input) and **returns** a string with the normalized name: it should include the first and last name (as in the previous exercise) and abbreviations of all other intermediate names, such as in the example bellow:
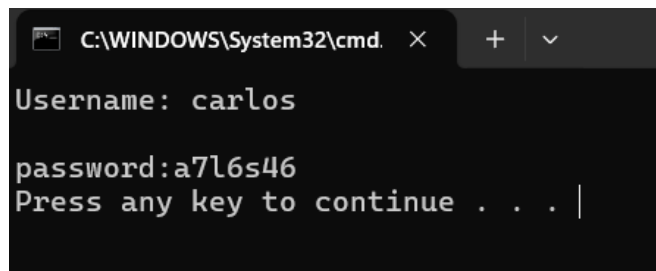
   Example:
   standardName('Carlos Alberto Costa Pereira') => Carlos A. C. Pereira

7. Implement the **generatePassword**(*userName*) function that works as a password generator: the function must **receive** a username, and based on that
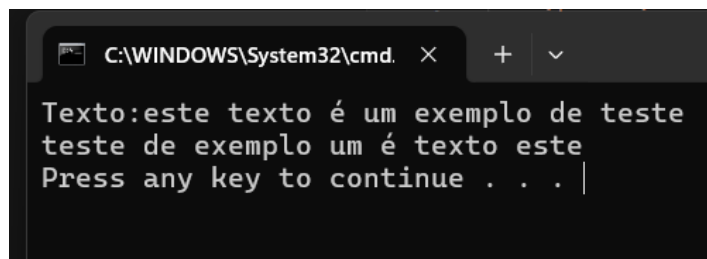
name it must generate and **return** a password which is generated as follows:

- The password consists of the characters from the **even positions of the username,** interspersed by a random number between 1 and 9 (in-clusive).

- The password **ends** with the number of characters included in the username.

- If the userName includes any spaces, the function should return the message "username is invalid" as an alternative to the pass-word.
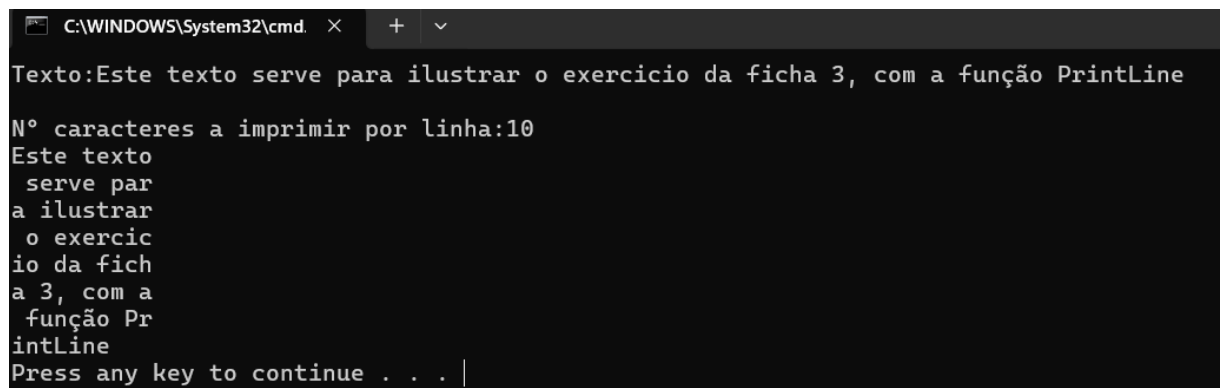
8. Write the function **reverseWords**(*text*) that **receives** a text and returns the same text, but with the words in reverse order.

9. Implement the function **printCharLine**(*text,numberChar*) that **receives two arguments:** a text, and the number of characters you want to print per line.

   Your function should print the text based on this number of characters, as shown in the image below.