

EjerciciosListas

March 13, 2024

Before you turn this problem in, make sure everything runs as expected. First, **restart the kernel** (in the menubar, select Kernel→Restart) and then **run all cells** (in the menubar, select Cell→Run All).

Make sure you fill in any place that says YOUR CODE HERE or “YOUR ANSWER HERE”, as well as your name and collaborators below:

```
[ ]: NAME = ""
COLLABORATORS = ""
```

Listas

Profesor: M. en C. Miguel Angel Pérez León

0.1 Wordplay

Constestar los siguientes ejercicios, para ser acredores a los 2 puntos sobre el primer parcial: - Debe entregarse en parejas - Todos los ejercicios deben pasar las pruebas - en caso de cración de funciones, comentarlas debidamente como como se mostró en clases:

```
def nombre_funcion(arg_1:tipo_dato, arg_2:tipo_dato,...) -> (tipo de dato de vuelta de la fun
    '''descripción de lo que hace la función
        - tipo de dato de entrada y valores que acepta
        - tipo de dato de salida y valores queu acepta

        poner un ejemplo de lo que hace la funcion
    >>>nombre_funcion(5,3)
    [out] 8
    ...
```

Todo inciso cuenta con su respectiva validación, **no continuar con incisos que no hayan pasado la validación.**

- (a) Todas las palabras que terminen con *ime*.
- (b) Cuantas palabras contienen al menos una de las siguientesletras *r, s, t, l, n, e*.
- (c) Porcentaje de palabras contienen al menos una de las siguientesletras *r, s, t, l, n, e*.
- (d) Todas las palabras que no contienen vocales (sin ninguna vocal) :).
- (e) Todas las palabras que contiene todas las vocales.

- (f) Responder si hay más palabras con 10 letras o con 7 letras.
- (g) La palabras más larga en la lista.
- (h) Todos los palindromos (un palindromo es una palabra que se lee igual de izquierda a derecha que viceversa).
- (j) Todas las palabras que contiene una q sin que sea seguida de una u .
- (k) Todas las palabras que contiene zu en cualquier parte de la palabra.
- (l) Todas las palabras que contienen una z y una w .
- (m) Todas las palabras cuya primer letra es una a , tercer letra es una e y quinta letra es una i .
- (n) Todas las palabras de 2 letras.
- (o) Todas las palabras de 4 letras que comienzan y terminan con la misma letra.
- (p) Cuantas vocales contiene una palabra.
- (q) Todas las palabras que contienen al menos 9 vocales.

0.2 Las siguientes lineas son de apoyo para que ustedes sólo procesen los ejercicios

Es necesario ejecutar las siguientes celdas, de otra forma no podrás realizar los ejercicios.

```
[ ]: from macti.evaluation import Quiz
quiz = Quiz("1", 'ManejoDatos', 'PrimerParcial', "local")
```

```
[ ]: !wget https://raw.githubusercontent.com/jugernaut/MexSIAM/main/Ejercicios/utils/
↪data/wordsInEnglish.txt
```

```
[ ]: import numpy as np

file = open('wordsInEnglish.txt', 'r') #abrir el txt
words = file.read() # lectura de txt y guardado en la variable words
file.close() # cerrar el txt
words[:100]

words = words.replace('\n', ', ')
```

```
[ ]: '''
LA VARIABLE WORDS CONTIENE LA LISTA DE PALABRAS, ALGO ASÍ
a
aa
aaa
aah
aahed
aahing
```

```

aahs
aal
aalii
aaliis
aals
aam
aani
aardvark
aardvarks
aardwolf
aardwolves
aargh
aaron
aaronic
aaronical
aaronite
aaronitic
aarrgh
'''
words = words.split(', ')

```

```

[ ]: # UTILIZA LA VARIABLE words para todos los incisos
len(words)

```

0.3 Consideraciones

- En cada inciso debes completar el código respetando la variable de retorno
- Respetar el orden alfabético.
- Cada inciso contiene su respectiva validación, contesta en orden, no avances si no has resuelto un inciso previo.

```

[ ]: # (a) All words ending in ime.
def termina_ime():
    # variables auxiliares
    end_ime = []
    count_ime = 0
    # YOUR CODE HERE
    raise NotImplementedError()
    return end_ime

```

```

[ ]: """
Evaluacion del inciso (a)
"""
file_answer.write('1', termina_ime(), 'Revisa el mensaje de error')

```

```

[ ]: # (b) How many words contain at least one of the letters _r, s, t, l, n, e_.
def cuantas_contienen():

```

```

# variables auxiliares
counter = 0
letters = ['r', 's', 't', 'l', 'n', 'e']
# YOUR CODE HERE
raise NotImplementedError()
return counter

```

```

[ ]: file_answer.write('2', np.array(cuantas_contienen()), 'Revisa el mensaje de_
↳error')

```

```

[ ]: # (c) The percentage of words that contain at least one of the letters _r, s,
↳t, l, n, e_.
def porcentaje():
    # variables auxiliares
    por = 0
    # YOUR CODE HERE
    raise NotImplementedError()
    return por

```

```

[ ]: file_answer.write('3', np.array(porcentaje()), 'Revisa el mensaje de error')

```

```

[ ]: # (d) All words with no vowels.
def cuantas_sin_vocales():
    vocales = 'aeiou'
    con_vocales = 0
    # YOUR CODE HERE
    raise NotImplementedError()
    return sin_vocales

```

```

[ ]: file_answer.write('4', np.array(cuantas_sin_vocales()), 'Revisa el mensaje de_
↳error')

```

```

[ ]: # (e) All words that contain every vowel.
def palabras_con_vocales():
    vowels = set("aeiou")
    pal_vocales = 0
    # YOUR CODE HERE
    raise NotImplementedError()
    return pal_vocales

```

```

[ ]: file_answer.write('5', np.array(palabras_con_vocales()), 'Revisa el mensaje de_
↳error')

```

```

[ ]: # (f) Whether there are more ten-letter words or seven-letter words.

# Cuenta cuantas palabras hat con 10 letras y cuantas hay con 7 y muestra el_
↳resultado

```

```
# YOUR CODE HERE
raise NotImplementedError()
```

```
[ ]: # (g) The longest word in the list.
def palabra_mas_larga():
    palabra_larga = ""
    # YOUR CODE HERE
    raise NotImplementedError()
    return palabra_larga
```

```
[ ]: file_answer.write('6', np.array(palabra_mas_larga()), 'Revisa el mensaje de_
↳error')
```

```
[ ]: # (h) All palindromes.
def todos_palindromos():
    palim_list = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return palim_list
```

```
[ ]: file_answer.write('7', np.array(todos_palindromos()), 'Revisa el mensaje de_
↳error')
```

```
[ ]: # (j) All words that contain a "q" that isn't followed by a "u".
def contienen_q():
    lista_q = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_q
```

```
[ ]: file_answer.write('8', np.array(contienenen_q()), 'Revisa el mensaje de error')
```

```
[ ]: # (k) All words that contain "zu" anywhere in the word.
def contienen_zu():
    lista_zu = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_zu
```

```
[ ]: file_answer.write('9', np.array(contienenen_zu()), 'Revisa el mensaje de error')
```

```
[ ]: # (l) All words that contain both a "z" and a "w".
def contienen_zw():
    lista_zw = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_zw
```

```
[ ]: file_answer.write('10', np.array(contienen_zw()), 'Revisa el mensaje de error')
```

```
[ ]: # (m) All words whose first letter is "a", third letter is "e" and fifth letter
      ↪ is "i".
def contienen_aei():
    lista_aei = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_aei
```

```
[ ]: file_answer.write('11', np.array(contienen_aei()), 'Revisa el mensaje de error')
```

```
[ ]: # (n) All two-letter words.
def palabras_dea_dos():
    lista_doble = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_doble
```

```
[ ]: file_answer.write('12', np.array(palabras_dea_dos()), 'Revisa el mensaje de
      ↪error')
```

```
[ ]: # (o) All four-letter words that start and end with the same letter.
def inicio_fin():
    lista_if = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return lista_if
```

```
[ ]: file_answer.write('13', np.array(inicio_fin()), 'Revisa el mensaje de error')
```

```
[ ]: # (p) Define una funcion que recib una palabra y cuente el numero de vocales
def cuenta_vocales(palabra):
    vocales = 'aeiou'
    vocales_palabras = []
    # YOUR CODE HERE
    raise NotImplementedError()
    return cuantas_vocales
```

```
[ ]: file_answer.write('14', np.array(cuenta_vocales('por eso guero')), 'Revisa el
      ↪mensaje de error')
```

0.3.1 HINT PUEDES USAR FUNCIONES PREVIAS

```
[ ]: # (q) All words that contain at least nine vowels.  
def nueve_vocales():  
    vocales = 0  
    p_list = []  
    # YOUR CODE HERE  
    raise NotImplementedError()  
    return vocales
```

```
[ ]: file_answer.write('15', np.array(nueve_vocales()), 'Revisa el mensaje de error')
```

```
[ ]: file_answer.to_file('1')
```