

Victor Hugo Enriquez Gonzalez 0210389

1 - Resuelve los siguientes ejercicios utilizando el mapa de nido a continuación.

A) - Crea un mapa que contenga los datos del archivo paises.csv en el año 2013

1 Clave: El nombre del país

2 Valor: La expectativa de vida

```
In [53]: import csv
from mapas import mapa

file=open('países.csv')
lector=csv.reader(file)
mapaTodos=mapa()

print('----Lista de los datos del archivo países.csv en el año 2013')
lista=[x for x in lector if x[2]=='2013']
for elemento in lista:

    lugar=str(elemento[0])
    expectativa=float(elemento[1])
    mapaTodos[lugar]=expectativa
    print(mapaTodos.getKey(lugar) + " = \t " +str( mapaTodos[lugar]))
```

```

----Lista de los datos del archivo paises.csv en el año 2013
Canada=      81.2380487804878
Sao Tome and Principe=  66.1348780487805
Lao PDR=      67.8059268292683
Arab World=    69.9497024315618
Latin America & Caribbean (all income levels)=  74.6386538292174
Cambodia=      71.4088292682927
Ethiopia=      62.9659512195122
Aruba=  75.206756097561
Sri Lanka=      74.0680487804878
Swaziland=      48.8506341463415
South Asia=      66.6447069079113
Argentina=      76.0126829268293
Bolivia=      66.9266341463415
Bahamas, The=    74.9144634146342
Burkina Faso=    55.8620243902439
Bahrain=      76.5361951219512
Saudi Arabia=    75.4969024390244
Middle East & North Africa (all income levels)=  72.18730285988
78
Guatemala=      71.6638536585366
Guinea=      55.8446585365854
World=  70.7792433488252
St. Lucia=      74.6741707317073
Congo, Rep.=    58.2965609756098
Spain=  82.3780487804878
Liberia=      60.2063902439024
Maldives=      77.5738292682927
East Asia & Pacific (all income levels)=  74.7328719064231
Oman=  76.5910487804878
Tanzania=      60.8464390243903
Cabo Verde=      74.5437317073171
Gabon=  63.0739268292683
European Union=  80.5444282551629
Jamaica=      73.2824878048781
Albania=      77.3504634146341
United Arab Emirates=  76.9578780487805
Guam=  78.6185609756098
India=  66.2108536585366
Madagascar=    64.2486341463415
Lesotho=      48.836
OECD members=  79.9551297376392
Pacific island small states=  69.6833906137897
Turkey=      74.8624390243903
Virgin Islands (U.S.)=  79.4731707317073
Venezuela, RB=  74.487512195122
Bangladesh=    70.2948536585366
Solomon Islands=  67.5070731707317
Lebanon=      79.8463658536585
Mongolia=      67.3394390243902
France=      82.5658536585366
Syrian Arab Republic=  74.7107073170732
Rwanda=      63.4928536585366
Somalia=      54.691243902439
Peru=  74.5155365853659
Vanuatu=      71.4106829268293
Cote d'Ivoire=  50.4017317073171

```

Benin= 59.1194878048781
 Other small states= 59.5348700989316
 Cuba= 79.0670243902439
 Cameroon= 54.587512195122
 Low & middle income= 68.8534304427285
 Togo= 56.1503414634146
 China= 75.199512195122
 Dominican Republic= 73.2329756097561
 Germany= 80.8926829268293
 Ghana= 60.9471219512195
 Tonga= 72.489
 Indonesia= 70.607243902439
 Latin America & Caribbean (developing only)= 74.4619940680238
 Colombia= 73.7770731707317
 Libya= 75.1760487804878
 Finland= 80.6268292682927
 Central African Republic= 49.4753902439024
 North America= 78.9904004653318
 Vietnam= 75.6066829268293
 Kenya= 61.0831707317073
 Korea, Dem. Rep.= 69.5006341463415
 Bulgaria= 74.3146341463415
 Mauritius= 73.5663414634146
 Romania= 74.5634146341464
 Angola= 51.464
 Central Europe and the Baltics= 76.0458627253259
 Egypt, Arab Rep.= 70.907243902439
 South Africa= 56.0983170731707
 St. Vincent and the Grenadines= 72.4012195121951
 Fiji= 69.7447317073171
 Caribbean small states= 71.7986120967827
 Brunei Darussalam= 78.3734634146341
 Middle income= 70.0756180569818
 Austria= 80.9365853658537
 High income: OECD= 80.6851191411459
 Mozambique= 49.8362682926829
 Uganda= 58.6479756097561
 Japan= 83.0960975609756
 Niger= 57.9661707317073
 United States= 78.7414634146342
 Afghanistan= 60.5091219512195
 Kuwait= 74.359
 Panama= 77.3680975609756
 Mali= 54.6039268292683
 Costa Rica= 79.7050243902439
 Euro area= 81.61444514232
 Ireland= 80.8951219512195
 Pakistan= 66.4358780487805
 Nigeria= 52.1090243902439
 Lower middle income= 66.2090244490233
 Ecuador= 76.1925609756098
 Australia= 82.0951219512195
 Algeria= 70.8821707317073
 Chile= 79.5726585365854
 Puerto Rico= 78.5355365853659
 Belize= 73.6991463414634
 Fragile and conflict affected situations= 59.9591409658314

Sierra Leone= 45.3290487804878
 East Asia & Pacific (developing only)= 73.8222534142487
 Denmark= 80.0512195121951
 Philippines= 68.5537317073171
 Portugal= 80.3731707317073
 Morocco= 70.6431219512195
 Namibia= 63.8811463414634
 French Polynesia= 76.1290243902439
 Guinea-Bissau= 54.0336341463415
 Kiribati= 68.5323414634147
 Switzerland= 82.6975609756098
 Grenada= 72.6103414634146
 Chad= 50.7005853658537
 Uruguay= 76.9078292682927
 Equatorial Guinea= 52.6129024390244
 Korea, Rep.= 81.3682926829268
 Djibouti= 61.3039756097561
 Antigua and Barbuda= 75.6653170731707
 Low income= 61.6187247095495
 High income= 79.2952747565341
 Burundi= 53.628756097561
 Least developed countries: UN classification= 61.1251710282053
 Cyprus= 79.6371219512195
 Barbados= 75.1320243902439
 Qatar= 78.4538048780488
 Italy= 82.9365853658537
 Sudan= 61.8639512195122
 Sub-Saharan Africa (developing only)= 56.4419985010462
 Singapore= 82.1439024390244
 Malta= 80.7463414634146
 Netherlands= 81.1048780487805
 Suriname= 70.8113902439024
 Iran, Islamic Rep.= 73.7627317073171
 Israel= 81.7048780487805
 Malaysia= 74.8422926829268
 Iceland= 82.9170731707317
 Zambia= 57.0225853658537
 Sub-Saharan Africa (all income levels)= 56.4389078290904
 Senegal= 63.2021707317073
 Papua New Guinea= 62.2989268292683
 Malawi= 54.7237804878049
 Zimbabwe= 58.0459756097561
 Jordan= 73.7473902439024
 Poland= 76.7975609756098
 Mauritania= 61.3506097560976
 Small states= 63.2030657750453
 New Caledonia= 76.2534146341464
 Trinidad and Tobago= 69.8171463414634
 Hungary= 75.0634146341463
 Honduras= 73.4934390243903
 Myanmar= 64.9324634146342
 Mexico= 77.1350731707317
 Tunisia= 75.1
 Nicaragua= 74.4653902439024
 Congo, Dem. Rep.= 49.6232682926829
 Comoros= 60.644
 United Kingdom= 81.5

```
Gambia, The=      58.6070731707317
High income: nonOECD=      73.4428246456423
Greece=          80.6341463414634
Paraguay=       72.1937317073171
Botswana=       46.9907073170732
Heavily indebted poor countries (HIPC)=      58.2408523436924
```

B) - Obtén y muestra la expectativa de vida en Mexico, Guatemala, Belize, El Salvador y Nicaragua.

```
In [54]: file=open('países.csv')
lector=csv.reader(file)

mapaTodos=mapa()
listaTodos=[x for x in lector]
for elemento in listaTodos:

    lugar=str(elemento[0])
    expectativa=float(elemento[1])
    mapaTodos[lugar]=expectativa

print('Mexico : ' , mapaTodos['Mexico'])
print('Guatemala : ',mapaTodos['Guatemala'])
print('Belize : ' , mapaTodos['Belize'])
print('El Salvador : ' , mapaTodos['El Salvador'])
print('Nicaragua : ' , mapaTodos['Nicaragua'])
```

```
Mexico : 77.1350731707317
Guatemala : 71.6638536585366
Belize : 73.6991463414634
El Salvador : 67.5016097560976
Nicaragua : 74.4653902439024
```

C) - Encuentra la expectativa de vida para cada uno de los países anteriores.

```
In [55]: print ("País: {0} \t\t\t Expectativa: {1}".format("Mexico",mapaTodos['Mexico']))
print ("País: {0} \t\t\t Expectativa: {1}".format("Guatemala",mapaTodos['Guatemala']))
print ("País: {0} \t\t\t Expectativa: {1}".format("Belize",mapaTodos['Belize']))
print ("País: {0} \t\t\t Expectativa: {1}".format("El Salvador",mapaTodos['El Salvador']))
print ("País: {0} \t\t\t Expectativa: {1}".format("Nicaragua",mapaTodos['Nicaragua']))
```

```
País: Mexico          Expectativa: 77.1350731707317
País: Guatemala       Expectativa: 71.6638536585366
País: Belize          Expectativa: 73.6991463414634
País: El Salvador     Expectativa: 67.5016097560976
País: Nicaragua       Expectativa: 74.4653902439024
```

D) - Obten la lista de todas las claves del mapa.

```
In [56]: file=open('países.csv')
lector=csv.reader(file)

#mapaTodos=mapa() <---Esta activado arriba
listaTodos=[x for x in lector]

for elemento in listaTodos:

#----Se eliminaron los repetidos
    if elemento[0] != lugar: #<-----NOTA: SI QUIERE MOSTRAR TODOS, ELI
MINAR ESTA LINEA
#----Todas las claves
        lugar=str(elemento[0])
        #----Imprime el key
        print(mapaTodos.getKey(lugar))
```

Canada
Sao Tome and Principe
Lao PDR
Arab World
Latin America & Caribbean (all income levels)
Cambodia
Ethiopia
Aruba
Sri Lanka
Swaziland
South Asia
Argentina
Bolivia
Bahamas, The
Burkina Faso
Bahrain
Saudi Arabia
Middle East & North Africa (all income levels)
Guatemala
Guinea
World
St. Lucia
Congo, Rep.
Spain
Liberia
Maldives
East Asia & Pacific (all income levels)
Oman
Tanzania
Cabo Verde
Gabon
New Zealand
European Union
Jamaica
Albania
United Arab Emirates
Guam
India
Madagascar
Lesotho
OECD members
Upper middle income
Pacific island small states
Turkey
Virgin Islands (U.S.)
Venezuela, RB
Bangladesh
Solomon Islands
Lebanon
Mongolia
France
Syrian Arab Republic
Rwanda
Somalia
Peru
Vanuatu
Norway

Cote d'Ivoire
Benin
Other small states
Cuba
Cameroon
Low & middle income
Togo
China
Dominican Republic
Germany
Ghana
Tonga
Indonesia
Latin America & Caribbean (developing only)
Colombia
Libya
Finland
Central African Republic
North America
Sweden
Vietnam
Guyana
Kenya
Korea, Dem. Rep.
Bulgaria
Mauritius
Romania
Angola
Central Europe and the Baltics
Egypt, Arab Rep.
South Africa
St. Vincent and the Grenadines
Fiji
Caribbean small states
Brunei Darussalam
Middle income
Austria
High income: OECD
Mozambique
Uganda
Japan
Niger
United States
Brazil
Afghanistan
Kuwait
Panama
Mali
Costa Rica
Euro area
Ireland
Pakistan
Nigeria
Lower middle income
Ecuador
Australia
Algeria

El Salvador
Chile
Puerto Rico
Thailand
Haiti
Belize
Fragile and conflict affected situations
Sierra Leone
Nepal
East Asia & Pacific (developing only)
Denmark
Philippines
Portugal
Morocco
Namibia
French Polynesia
Guinea-Bissau
Kiribati
Switzerland
Grenada
Yemen, Rep.
Iraq
Chad
Uruguay
Equatorial Guinea
Korea, Rep.
Djibouti
Antigua and Barbuda
Low income
High income
Burundi
Least developed countries: UN classification
Cyprus
Barbados
Qatar
Italy
Bhutan
Sudan
Sub-Saharan Africa (developing only)
Singapore
Malta
Netherlands
Suriname
Middle East & North Africa (developing only)
Iran, Islamic Rep.
Israel
Malaysia
Iceland
Zambia
Sub-Saharan Africa (all income levels)
Senegal
Papua New Guinea
Malawi
Zimbabwe
Jordan
Poland
Mauritania

Small states
 New Caledonia
 Trinidad and Tobago
 Hungary
 Honduras
 Myanmar
 Mexico
 Tunisia
 Nicaragua
 Congo, Dem. Rep.
 Comoros
 United Kingdom
 Gambia, The
 High income: nonOECD
 Greece
 Paraguay
 Botswana
 Heavily indebted poor countries (HIPC)

E) - Filtra la lista anterior para mostrar sólo los países que empiezan con "C"

```
In [57]: file=open('países.csv')
lector=csv.reader(file)

#mapaTodos=mapa() <---Esta activado arriba
listaTodos=[x for x in lector]
for elemento in listaTodos:

    cadena = elemento[0]
    #----Se eliminaron los repetidos
    if elemento[0] != lugar:
        if cadena[0] == 'C':
            lugar=str(elemento[0])
            print(mapaTodos.getKey(lugar))
```

Canada
 Cambodia
 Congo, Rep.
 Cabo Verde
 Cote d'Ivoire
 Cuba
 Cameroon
 China
 Colombia
 Central African Republic
 Central Europe and the Baltics
 Caribbean small states
 Costa Rica
 Chile
 Chad
 Cyprus
 Congo, Dem. Rep.
 Comoros

F) - Calcula el promedio de expectativa de vida para todos los paises que empiezan con C.

```
In [58]: file=open('paises.csv')
lector=csv.reader(file)

totalPaises=0
promedio=0
mapaTodos=mapa()
listaTodos=[x for x in lector]

for elemento in listaTodos:
    cadena = elemento[0]

    #-----Solo los paises que comienzan con 'C'
    if cadena[0] == 'C':

        lugar=str(elemento[0])
        expectativa=float(elemento[1])
        mapaTodos[lugar]=expectativa
        promedio = promedio + expectativa
        totalPaises = totalPaises + 1

promedio = promedio / totalPaises
print("Promedio de expectativa de vida = " + str(promedio))
```

Promedio de expectativa de vida = 63.60198347876371

2 - Resuelve los ejercicios utilizando el mapa definido a continuación

Feeling of Power
Isaac Asimov

Jehan Shuman was used to dealing with the men in authority on long-embattled earth. He was only a civilian but he originated programming patterns that resulted in self-directing war computers of the highest sort. Generals, consequently listened to him. Heads of congressional committees too. There was one of each in the special lounge of New Pentagon. General Weider was space-burned and had a small mouth puckered almost into a cipher. He smoked Denebian tobacco with the air of one whose patriotism was so notorious, he could be allowed such liberties. Shuman, tall, distinguished, and Programmer-first-class, faced them fearlessly. He said, "This, gentlemen, is Myron Aub." "The one with the unusual gift that you discovered quite by accident," said Congressman Brant placidly. "Ah." He inspected the little man with the egg-bald head with amiable curiosity. The little man, in return, twisted the fingers of his hands anxiously. He had never been near such great men before. He was only an aging low-grade technician who had long ago failed all tests designed to smoke out the gifted ones among mankind and had settled into the rut of unskilled labor. There was just this hobby of his that the great Programmer had found out about and was now making such a frightening fuss over. General Weider said, "I find this atmosphere of mystery childish." "You won't in a moment," said Shuman. "This is not something we can leak to the firstcomer. Aub!" There was something imperative about his manner of biting off that one-syllable name, but then he was a great Programmer speaking to a mere technician. "Aub! How much is nine times seven?" Aub hesitated a moment. His pale eyes glimmered with a feeble anxiety. "Sixty-three," he said. Congressman Brant lifted his eyebrows. "Is that right?" "Check it for yourself, Congressman." The congressman took out his pocket computer, nudged the milled edges twice, looked at its face as it lay there in the palm of his hand, and put it back. He said, "Is this the gift you brought us here to demonstrate. An illusionist?" "More than that, sir. Aub has memorized a few operations and with them he computes on paper." "A paper computer?" said the general. He looked pained.

A) - Crea un mapa, analizando el texto al inicio de este ejercicio, que tenga como valores:

1 Clave: Una letra minúscula

2 Valor: Las veces que aparece esa letra, ya sea mayúscula o minúscula en el texto

```
In [59]: file=open('texto.txt')
lector=file.read().lower()  #-<<<---Se hace todo el texto en minuscula p
ara detectar todas las letras

mapaTexto=mapa()
abc="abcdefghijklmnopqrstuvwxyz"

listaTexto=[x for x in lector]
listaAbc=[x for x in abc]
listaCont=[x for x in range(0,27)]

for x in listaCont:
    listaCont[x]=0

aux=0
for letra in listaAbc:
    for elemento in listaTexto:
        if letra==elemento:
            listaCont[aux]=listaCont[aux] + 1
        aux=aux+1

aux=0
for elemento in listaAbc:

    abc=str(elemento)
    repetido=int(listaCont[aux])
    mapaTexto[abc]=repetido
    print ("Letra: {0} \t Numero de veces repetida: {1}".format(mapaText
o.getKey(abc),str(mapaTexto[abc])))
    aux=aux+1
```

Letra: a	Numero de veces repetida: 152
Letra: b	Numero de veces repetida: 31
Letra: c	Numero de veces repetida: 47
Letra: d	Numero de veces repetida: 70
Letra: e	Numero de veces repetida: 221
Letra: f	Numero de veces repetida: 36
Letra: g	Numero de veces repetida: 52
Letra: h	Numero de veces repetida: 111
Letra: i	Numero de veces repetida: 133
Letra: j	Numero de veces repetida: 2
Letra: k	Numero de veces repetida: 14
Letra: l	Numero de veces repetida: 68
Letra: m	Numero de veces repetida: 65
Letra: n	Numero de veces repetida: 123
Letra: ñ	Numero de veces repetida: 0
Letra: o	Numero de veces repetida: 119
Letra: p	Numero de veces repetida: 31
Letra: q	Numero de veces repetida: 2
Letra: r	Numero de veces repetida: 88
Letra: s	Numero de veces repetida: 125
Letra: t	Numero de veces repetida: 160
Letra: u	Numero de veces repetida: 55
Letra: v	Numero de veces repetida: 7
Letra: w	Numero de veces repetida: 34
Letra: x	Numero de veces repetida: 3
Letra: y	Numero de veces repetida: 23
Letra: z	Numero de veces repetida: 1

B) - Imprime cuántas vocales hay en el texto

```
In [60]: aux=0
for elemento in listaTexto:
    if elemento == 'a' or elemento == 'e' or elemento == 'i' or elemento
    == 'o' or elemento == 'u':
        aux=aux+1

print ("Numero de vocales en el texto : " + str(aux))
```

Numero de vocales en el texto : 680

C) - Imprime cada pareja del mapa, en forma clave:valor

```
In [61]: aux=0
        for elemento in listaAbc:

            abc=str(elemento)
            print ("Clave: {0} \t Valor: {1}".format(mapaTexto.getKey(abc),str(m
            apaTexto[abc])))
            aux=aux+1
```

Clave: a	Valor: 152
Clave: b	Valor: 31
Clave: c	Valor: 47
Clave: d	Valor: 70
Clave: e	Valor: 221
Clave: f	Valor: 36
Clave: g	Valor: 52
Clave: h	Valor: 111
Clave: i	Valor: 133
Clave: j	Valor: 2
Clave: k	Valor: 14
Clave: l	Valor: 68
Clave: m	Valor: 65
Clave: n	Valor: 123
Clave: ñ	Valor: 0
Clave: o	Valor: 119
Clave: p	Valor: 31
Clave: q	Valor: 2
Clave: r	Valor: 88
Clave: s	Valor: 125
Clave: t	Valor: 160
Clave: u	Valor: 55
Clave: v	Valor: 7
Clave: w	Valor: 34
Clave: x	Valor: 3
Clave: y	Valor: 23
Clave: z	Valor: 1

D) - Imprime una lista que contenga, para cada letra de la a a la z, si está o no está en el texto.

Ejemplo: a: sí


```
In [62]: aux=0
for elemento in listaAbc:

    abc=str(elemento)
    if mapaTexto[abc] != 0:
        print(mapaTexto.getKey(abc) + " : si")
    else:
        print(mapaTexto.getKey(abc) + " : no")
    aux=aux+1
```

```
a : si
b : si
c : si
d : si
e : si
f : si
g : si
h : si
i : si
j : si
k : si
l : si
m : si
n : si
ñ : no
o : si
p : si
q : si
r : si
s : si
t : si
u : si
v : si
w : si
x : si
y : si
z : si
```