Programación—C.Recuperativo - Martes 06 de Diciembre de 2016

Nombre

Rol — Paralelo

1. [20 %] Realice el ruteo de los siguientes programas e indique qué es lo que imprimen. Cada vez que el valor de una variable cambie, escríbalo en una nueva fila de la tabla. Recuerde que si una variable es de tipo string, debe colocar su valor entre comillas simples ' '. Si una variable almacena una función, coloque el nombre de ésta como valor (sin comillas).

***Importante**: La tabla tiene suficientes filas.*

```
def valor(c):
    cod, cnt = 'NCRAV', 0
    while cnt < len(cod):
        if cod[cnt] == c:
            return cnt
        cnt += 1

R = 'RRC'
D = valor(R[0])
D += 10 * valor(R[1])
D += 10 ** valor(R[2])
print D
```

global		valor		
R	D	c	cod	cnt

Complete el Algoritmo

A partir de los ejemplos de ejecución de la siguiente función, complete el código indicando el número de la línea que corresponde insertar a partir de las opciones presentadas en el cuadrado de la derecha **(Opciones)**. Notar que en este cuadrado hay más líneas de las que se requieren para resolver el problema y que las líneas de código que ya existen en la parte izquierda, no se pueden borrar o correr.

```
...(8).... # ejemplo

    i = hora.index(':')

    .........

    .........

    resultado += ','

    .........

    return resultado
```

Opciones:

```
(1)     i = 0
(2)     resultado = 0
(3)     resultado = hora[:i+j]
(4)     j = hora[i:].index(':')
(5)     j = hora[i+1:].index(':')
(6)     resultado += hora[i+j+2:]
(7)     resultado += hora[i+j+1:]
(8)     def notacion(hora):
(9)     resultado = hora[:i+j+1]
```

```
>>> notacion('19:20:22')
'19:20,22'
>>> notacion('08:17:04')
'08:17,04'
```

Programación—C.Recuperativo - Martes 06 de Diciembre de 2016

Nombre

Rol — Paralelo

2. **[40 %]** La Oficina de Asuntos Internacionales (OAI) le ha pedido que construya un programa para ”traducir´´ ciertas palabras del español coloquial ”CL´´ al español coloquial de otros países. Para esto dispone de un diccionario que tiene por llave el término en CL y como valores, una lista de tuplas de equivalencias por país, notar que si para un país en particular no existe la tupla, es porque la palabra se llama de la misma forma que en CL. Además, se entrega un diccionario con los códigos y nombres de países.

```
terminos_cl = {
    'curado': [('ec', 'chumado'), ('ve', 'borracho')],
    'cabro': [('ec', 'joven'), ('ve', 'chamo')],
    'polera':[('ec', 'camiseta'), ('pe', 'polo'), ('co', 'camiseta'), (
       've', 'camiseta')],
    'palta': [('ve', 'aguacate'), ('co', 'aguacate')], # ...
}

codigos = {'ec': 'Ecuador', 've': 'Venezuela', 'co': 'Colombia', 'pe':
   'Peru'}
```

Ahora se le solicita lo siguiente:

a) Escribir una función `traduce(terminos, texto, codigo)` que reciba el diccionario de `terminos`, un `texto` y un `codigo` de país. La función retorna el texto traducido al país solicitado. Las palabras que no tengan traducción para el idioma solicitado, se deben mantener idénticas. Asuma que **no** hay problemas de género y número.

```
>>> frase = 'El cabro curado se mancha la polera con palta'
>>> traduce(terminos_cl, frase, 've')
'El chamo borracho se mancha la camiseta con aguacate'
```

b) Escribir una función `traduce_todos(terminos, codigos, texto)` que reciba el diccionario de `terminos`, el diccionario de `codigos` y un `texto`. La función retorna un diccionario con el nombre del país como llave y el texto traducido como valor. Debe retornar todos los países disponibles en el diccionario de `codigos`.

```
>>> frase = 'El cabro curado se mancha la polera con palta'
>>> traduce_todos(terminos_cl, codigos, frase)
{'Colombia': 'El cabro curado se mancha la camiseta con aguacate',
'Ecuador': 'El joven chumado se mancha la camiseta con palta',
'Peru': 'El cabro curado se mancha la polo con palta',
'Venezuela': 'El chamo borracho se mancha la camiseta con aguacate'}
```

c) La OAI también desea identificar algunas palabras usadas en otros países. Para lo cual le pide escribir la función `buscar(terminos, codigos, palabra)` la cual recibe el diccionario de `terminos`, el diccionario de `codigos` y una `palabra`. La función retorna una tupla compuesta por la palabra traducida ”CL´´ y un conjunto de países que utilizan la palabra pasada por parámetro.

```
>>> buscar(terminos_cl, codigos, 'camiseta')
('polera', set(['Colombia', 'Ecuador', 'Venezuela']))
```

Programación—C.Recuperativo - Martes 06 de Diciembre de 2016

Nombre

Rol — Paralelo

3. [**40 %**] Las AFP administran los fondos de pensiones de todos los chilenos, salvo para el personal uniformado como ejército (capredena), carabineros (dipreca), etc.

Existe un archivo con la información previsional de todos los chilenos. Las líneas contienen el `rut/apellido/administradora` y en el caso de pertenecer a una rama uniformada además tiene la rama. Dos líneas de **ejemplo** se ven en el archivo `todos.txt`.

`todos.txt`

```
0-0/Pratt/Capredena/Armada
1-9/Perez/Habitat
```

Por otra parte se cuenta con un archivo que corresponde a las personas que se van a jubilar. En este archivo cada línea es del tipo `rut/monto_pension`. Una línea de este archivo se vería así: `0-0/450000`

Elabore la función `jubilar(Ageneral, Ajubilados, Asalida)` que reciba el nombre del archivo que contiene información previsional de todos los chilenos (`Ageneral`), el nombre del archivo que contiene las personas que se jubilarán este mes (`Ajubilados`) y el nombre del archivo (`Asalida`) donde se debe escribir el promedio de los montos de jubilación por cada rama de uniformados agrupadas por su administrador de pensión. Siga el formato del archivo **ejemplo** `salida.txt`. **Toda rama sin jubilados debe ser omitida en el archivo de salida**.

`salida.txt`

```
CAPREDENA
Ejercito 450000
Armada 500000

DIPRECA
Carabineros 300000
Gendarmeria 1400000
```