

Tarea UVA 7: Listas (Segunda Parte)

Esta parte debe entregarse a más tardar el sábado 3 de junio a las 23:55 horas



1. Implemente la función `clasificar(tiempos, n)`, que recibe como parámetro una lista con los tiempos obtenidos por los pilotos en las distintas vueltas de una sesión en particular, siguiendo el formato mostrado anteriormente, y un número entero que indica la cantidad de pilotos a clasificar a partir de esa lista. La función debe retornar una lista que contiene 2 listas ambas con el mismo formato. La primera lista corresponde una lista de listas con los n pilotos clasificados; cada piloto debe estar en una lista de dos elementos que contiene el menor tiempo y el nombre del piloto. La segunda lista corresponde al resto de los pilotos que no lograron clasificar, una lista por piloto, con el mejor tiempo y su nombre. Dependiendo de su desempeño, cada piloto debe aparecer una única vez, y sólo en una de las listas. Ambas listas deben estar ordenadas de menor a mayor, de acuerdo al tiempo.

Ejemplos

```
>>> tiempos = [['p5', 93526], ['p5', 91019], ['p8', 90207], ['p5', 94878], ['p6', 92827], ['p3', 93335], ['p4', 92681], ['p4', 90805], ['p10', 90825], ['p3', 90343], ['p5', 91890], ['p6', 90781], ['p3', 93618], ['p3', 91794]]
>>> print(clasificar(tiempos, 3))
[[[90207, 'p8'], [90343, 'p3'], [90781, 'p6']], [[90805, 'p4'], [90825, 'p10'], [91019, 'p5']]]
>>> print(clasificar(tiempos, 2))
[[[90207, 'p8'], [90343, 'p3']], [[90781, 'p6'], [90805, 'p4'], [90825, 'p10'], [91019, 'p5']]]
```

2. Implemente un programa que permita procesar los datos de una carrera. El programa debe hacer lo siguiente:

- a. Solicitar la lista que contiene los tiempos de la etapa de práctica (P).
- b. Utilizar la función `clasificar` para determinar los 10 pilotos que clasifican a Q2, y los que deben correr la Q1. En ambos casos debe mostrarse por pantalla los nombres de los pilotos y sus tiempos.
- c. Solicitar la lista que contiene los tiempos de la clasificación Q1.
- d. Utilizar la función `clasificar` para determinar los 2 pilotos que clasifican a Q2, mostrando en la pantalla sus nombres y tiempos.
- e. Solicitar la lista que contiene los tiempos de la clasificación Q2.
- f. Utilizar la función `clasificar` para determinar el orden de los primeros 12 pilotos en la línea de salida, y agregar desde el puesto 13 en adelante a los que no lograron clasificar en Q2, de acuerdo a sus tiempos en esa ronda de clasificación. El programa debe mostrar el orden final de los pilotos, pero únicamente sus nombres.

Para comprender a cabalidad el comportamiento que debe tener el programa, así como los formatos de entradas y salidas, guíese por el ejemplo.

Nota: Tome en cuenta que la función `input()` lee *strings*, por lo que para leer una lista y asignarla a una variable debemos utilizar la función `eval()`. Por ejemplo:

```
lista = eval(input())
```

Eso permite que ingresemos una lista de la forma `[...]` y se interprete correctamente.

Requisitos del programa

El programa y las funciones deben cumplir con los siguientes requisitos:

1. Puede utilizar **sólo la materia estudiada hasta la UVA 7** de IWI-131 Programación.
2. La tarea debe realizarse de manera **estrictamente individual**. No se arriesgue a graves sanciones.
3. Las función solicitada deben **cumplir fielmente** la especificación dada, incluyendo los parámetros y valores de retorno.
4. Tome en cuenta que **no sabemos de antemano cuántos pilotos participan, ni cuántas vueltas logra dar cada uno en cada etapa**. Lo único que se puede suponer es que siempre habrá al menos 15 pilotos.

Importante: Puede suponer que los datos ingresados son siempre correctos. No es necesario hacer validaciones ni considerar casos en que no se cumpla con lo especificado.

Ejemplo de ejecución:

A continuación se muestra una ejecución del programa que usted debe implementar. Se destacan en **negrita** los valores ingresados por el/la usuario/a. Observe que es una ejecución muy extensa.

```
Bienvenidos al Gran Premio de Pythonia (Pythonia Moto Grand Prix 2022)
```

```
Ingrese tiempos de la sesión de práctica: [['p3', 92045], ['p18', 92216],
['p1', 91781], ['p11', 91214], ['p3', 93744], ['p14', 92370], ['p16',
92552], ['p18', 90915], ['p6', 93665], ['p19', 93815], ['p2', 91307],
['p17', 90149], ['p11', 94899], ['p10', 90727], ['p9', 91181], ['p13',
92069], ['p5', 92052], ['p10', 92048], ['p14', 94682], ['p1', 94049], ['p1',
94560], ['p9', 92607], ['p16', 94602], ['p14', 91569], ['p14', 90344],
['p2', 92646], ['p8', 94626], ['p15', 93447], ['p18', 91796], ['p14',
94247], ['p12', 94819], ['p11', 94542], ['p6', 94856], ['p18', 92776],
['p12', 91010], ['p4', 92350], ['p13', 92753], ['p9', 94339], ['p4', 91167],
['p18', 94433], ['p18', 93003], ['p10', 94298], ['p19', 92478], ['p7',
90329], ['p13', 92631], ['p10', 92558], ['p3', 90239], ['p2', 92194],
['p13', 91775], ['p14', 93096], ['p11', 93109], ['p13', 91503], ['p5',
94608], ['p13', 93187], ['p10', 92365], ['p1', 94308], ['p13', 93869],
['p8', 91820], ['p10', 94802], ['p4', 93669], ['p3', 91616], ['p4', 91581],
['p15', 91617], ['p20', 92029], ['p3', 92160], ['p17', 94922], ['p20',
90196], ['p20', 94329], ['p11', 92476], ['p8', 94900], ['p4', 90146], ['p1',
91374], ['p9', 91783], ['p16', 93198], ['p17', 90648], ['p19', 92223],
['p6', 94163], ['p19', 90755], ['p6', 94704], ['p11', 94769], ['p2', 94227],
['p11', 94994], ['p7', 93577], ['p7', 93861], ['p12', 93236], ['p13',
90091], ['p12', 90252], ['p15', 94328], ['p3', 94337], ['p8', 92501], ['p2',
91143], ['p13', 93206], ['p5', 94920], ['p2', 90732], ['p12', 90135], ['p3',
92043], ['p6', 94034], ['p20', 92107], ['p5', 90626], ['p19', 92599],
['p19', 94474], ['p15', 94472], ['p5', 90231], ['p2', 93163], ['p16',
91210], ['p16', 90641], ['p6', 93027], ['p15', 91221], ['p16', 94986],
['p8', 94413], ['p19', 91988], ['p15', 91376], ['p8', 90076], ['p5', 92153],
['p20', 93629], ['p2', 92769], ['p7', 90395], ['p12', 90586], ['p2', 93378],
['p16', 91414], ['p9', 90441], ['p17', 90119], ['p17', 93521], ['p16',
91040], ['p16', 93131], ['p17', 92232], ['p5', 91121], ['p9', 90705], ['p7',
94309], ['p17', 91323], ['p7', 92804], ['p17', 94684], ['p6', 90393], ['p7',
94828], ['p2', 94706], ['p6', 90833], ['p20', 94534], ['p7', 90679], ['p8',
93992], ['p7', 91675], ['p7', 93115], ['p20', 93158], ['p20', 90497],
['p20', 94386], ['p6', 92196], ['p20', 93049]]
```

Clasifican directamente a Q2:

p8 con tiempo: 90076
p13 con tiempo: 90091
p17 con tiempo: 90119
p12 con tiempo: 90135
p4 con tiempo: 90146
p20 con tiempo: 90196
p5 con tiempo: 90231
p3 con tiempo: 90239
p7 con tiempo: 90329
p14 con tiempo: 90344

Ingrese tiempos de la sesión Q1: [['p9', 90597], ['p1', 90353], ['p2', 92841], ['p6', 90085], ['p15', 90608], ['p11', 91490], ['p11', 90794], ['p10', 91588], ['p11', 93268], ['p16', 92138], ['p1', 94796], ['p10', 93673], ['p15', 94935], ['p16', 90578], ['p1', 91886], ['p9', 90495], ['p19', 93563], ['p10', 90662], ['p18', 93848], ['p1', 90632], ['p1', 90256], ['p16', 94884], ['p11', 91078], ['p9', 91771], ['p19', 93271], ['p19', 91894], ['p1', 94879], ['p18', 92607], ['p1', 91805], ['p11', 90037], ['p1', 91436], ['p16', 92856], ['p9', 91345], ['p9', 93796], ['p15', 91499], ['p6', 90377], ['p15', 91004], ['p18', 91130], ['p18', 94853], ['p15', 92582], ['p2', 92774], ['p2', 94959], ['p16', 94834], ['p6', 92319], ['p9', 94097], ['p10', 93181], ['p9', 91859], ['p15', 92249], ['p19', 92051], ['p19', 91296], ['p18', 93528], ['p19', 93420], ['p10', 93234], ['p10', 94379], ['p1', 92354], ['p2', 90752], ['p6', 94335], ['p2', 90167], ['p19', 94053], ['p19', 90633], ['p18', 94165], ['p9', 94836], ['p15', 90741], ['p15', 93127], ['p6', 90065], ['p15', 93276], ['p18', 93617], ['p6', 94434], ['p18', 94451], ['p18', 90893], ['p6', 91603], ['p9', 92223], ['p10', 94225], ['p6', 90928], ['p9', 91163], ['p10', 92394], ['p6', 94651], ['p6', 90306]]

Clasifican a Q2

[[90037, 'p11'], [90065, 'p6']]

Orden de partida del puesto 13 en adelante:

[[90167, 'p2'], [90256, 'p1'], [90495, 'p9'], [90578, 'p16'], [90608, 'p15'], [90633, 'p19'], [90662, 'p10'], [90893, 'p18']]

Ingresa tiempos de la sesión Q2: [['p7', 90333], ['p4', 94513], ['p3', 90483], ['p6', 93319], ['p17', 92055], ['p13', 92890], ['p12', 93339], ['p4', 90752], ['p20', 93082], ['p5', 93323], ['p3', 94735], ['p20', 92974], ['p6', 94558], ['p17', 94964], ['p7', 93829], ['p5', 90260], ['p3', 94155], ['p4', 90529], ['p5', 94296], ['p14', 91194], ['p6', 92635], ['p14', 92429], ['p14', 94511], ['p13', 94854], ['p14', 90211], ['p6', 94948], ['p17', 90993], ['p13', 91925], ['p13', 92924], ['p3', 94933], ['p8', 91321], ['p11', 94204], ['p6', 91955], ['p11', 92711], ['p14', 94170], ['p11', 92713], ['p4', 90833], ['p5', 93296], ['p20', 93469], ['p17', 91311], ['p7', 92894], ['p11', 92631], ['p3', 90019], ['p8', 92295], ['p12', 94953], ['p17', 91185], ['p6', 90262], ['p6', 90432], ['p11', 92330], ['p12', 93342], ['p12', 90778], ['p3', 94083], ['p13', 90313], ['p7', 90895], ['p13', 90815], ['p20', 94804], ['p12', 92350], ['p12', 91838], ['p13', 92570], ['p13', 91078], ['p12', 93300], ['p12', 93556], ['p4', 93276], ['p11', 92495], ['p11', 92015], ['p4', 91200], ['p7', 92768], ['p4', 94888], ['p11', 91309], ['p13', 92770], ['p8', 90859], ['p8', 94599], ['p11', 90793], ['p8', 92348], ['p20', 94144], ['p5', 91061], ['p8', 92783], ['p13', 93755], ['p8', 94186], ['p20', 94878], ['p5', 92098], ['p7', 93084], ['p20', 91646], ['p5', 93219], ['p7', 91041], ['p5', 93655], ['p20', 90741], ['p20', 91713], ['p5', 90709], ['p5', 93139]]

Orden de partida para el Gran Premio de Pythonia 2022:

1. p3
2. p14
3. p5
4. p6
5. p13
6. p7
7. p4
8. p20
9. p12
10. p11
11. p8
12. p17
13. p2
14. p1
15. p9
16. p16
17. p15
18. p19
19. p10
20. p18