

Usted debe implementar dos funciones que me permitan mantener actualizada la información de una carrera de autos.

La primera función será la función `avances`.

1. `avances(novedades, historial_carrera)`

Esta función debe devolver una nueva lista con la cantidad de kilómetros que hizo cada auto hasta el momento, sumando las **novedades** del día actual al **historial_carrera**.

Argumentos de la función:

- **novedades:** Una lista que contiene la información de la carrera en el día analizado. Esta lista contiene pares de valores: el **número del auto** seguido de un **valor asociado** que puede ser:
 1. **Una cadena numérica de números enteros:** Representa la cantidad de kilómetros que ese auto ha recorrido.
 2. **Una letra (código):** Indica un evento relacionado con el auto. Los posibles códigos son:
 - **"A":** El auto abandonó la carrera.
 - **"P":** El auto terminó primero en ese día de la carrera.

Ejemplo:

`novedades = [1, "30", 2, "A", 3, "50", 4, "23", 3, "P"]`

Esta lista indica las siguientes actualizaciones:

- El auto **1** recorrió **30 km**.
- El auto **2** **abandonó** la carrera.
- El auto **3** recorrió **50 km**.
- El auto **4** recorrió **23 km**.
- El auto **3** terminó **primero** en ese día de carrera.
-
- **historial_carrera:** contiene la información actualizada hasta el momento de los kilómetros recorridos por cada auto. Esta lista contiene los pares de valores: el **número del auto** y el **total de kilómetros recorridos** por éste desde el inicio.

Ejemplo:

Si consideramos que el primer día de la carrera, la lista de novedades es la del ejemplo anterior y la del **historial_carrera** es vacía, entonces el nuevo historial de la carrera devuelto por la función será: `[1, 30, 3, 50, 4, 23]`

Debe tenerse en cuenta que el primer día de la competencia, **historial_carrera** es una lista vacía.

La lista devuelta por esta función debe tener un formato similar al de **historial_carrera** y **cumplir las mismas consideraciones de dicha lista** para permitir su uso en llamadas sucesivas.

Consideraciones:

- A. Tanto la lista **novedades** como la lista **historial_carrera** podrían estar vacías.
- B. Si las listas contienen información siempre tendrán un número par de elementos, **no es necesario validar esto**.
- C. Si un auto aparece más de una vez en la lista de **novedades**, todos los kilometrajes deben ser considerados y el avance es la suma de todos ellos.
- D. Puede suceder que un auto que abandona en la lista de **novedades**, aparezca primero o haya avanzado kilómetros. El abandono tiene prioridad sobre cualquier otro valor.
- E. La lista **novedades** no está necesariamente ordenada es decir que puede aparecer primero que un auto abandonó o es primero y luego sus kilómetros recorridos.
- F. El número del auto siempre será un entero positivo. **Debe validarse** solamente en la lista **novedades**.
- G. En la lista **novedades**, si el valor es una cadena numérica, representa una cantidad entera positiva de kilómetros recorridos. Si es un código, debe ser uno de los códigos válidos. **Debe validarse**.
- H. Si **historial_carrera** no es una lista vacía, y la identificación del auto en **novedades** no aparece en **historial_carrera**, no se debe considerar, por tanto no debe aparecer en el historial de la carrera devuelto.
- I. Si un auto que aparece en **historial_carrera** no aparece en las novedades, se considera que no avanzó, pero sigue en carrera (no abandonó).
- J. Cualquier entrada que deba validarse y resulte inválida no debe procesarse el par de valores, ni producirá interrupción en el programa, es decir, se debe ignorar. De ser inválidos todos los pares de valores, debe considerarse como una lista de novedades vacía.

2. top3(historial_carrera)

Esta función debe crear un archivo "**top3.txt**" que contenga como máximo los tres corredores con más kilómetros recorridos en total, ordenados según su posición en la carrera. Si el archivo ya existe, este se debe sobrescribir.

El formato para cada auto debe ser (ver ejemplos):

<nro de auto>: <cantidad de km> km

Argumentos de la función:

- **historial_carrera**: La lista acumulada devuelta por la función avances.

Consideraciones:

- En caso de empate, todos los empatados deben ocupar un lugar en el top 3. No importa el orden en el que aparecen.
- Si hubiera 2 corredores con 50 km, 2 corredores con 40km y 2 corredores con 30km, y el resto de los corredores con menos kilómetros el archivo deberá tener a los 6 corredores mencionados, ya que 2 ocupan el primer lugar, 2 el segundo y 2 el tercero.
- Si **historial_carrera** es una lista vacía, no se debe crear el archivo.

Ejemplo 1:

```
novedades = [1, "30", 2, "A", 3, "50", 4, "23", 3, "P"]
```

```
historial_carrera = avances(novedades, [])
```

Resultado:

```
historial_carrera = [1, 30, 3, 50, 4, 23]
```

```
novedades = [1, "50", 3, "20", 4, "60", 4, "P"]
```

```
historial_carrera = avances(novedades, historial_carrera)
```

Resultado:

```
historial_carrera = [1, 80, 3, 70, 4, 83]
```

```
top3(historial_carrera)
```

El contenido del archivo **"top3.txt"** debe ser:

4: 83 km

1: 80 km

3: 70 km

Ejemplo 2:

```
novedades1 = [2, "A", 1, "20", 2, "25", 1, "P", 4, "18"]
```

```
historial_carrera = avances(novedades1, [])
```

Resultado:

```
historial_carrera = [1, 20, 4, 18]
```

```
novedades2 = [1, "10", 4, "P", 4, "12"]
```

```
historial_carrera = avances(novedades2, historial_carrera)
```

Resultado:

```
historial_carrera = [1, 30, 4, 30]
```

```
top_3(historial_carrera)
```

El contenido del archivo **"top3.txt"** puede ser:

1: 30 km

4: 30 km

ó El contenido del archivo **"top3.txt"** puede ser:

4: 30 km

1: 30 km

Ejemplo 3:

Si **historial_carrera = [1, 30, 2, 10, 3, 30, 4, 45, 7, 5]**

top3(historial_carrera) generará el archivo con el siguiente contenido:

4: 45 km

1: 30 km

3: 30 km

2: 10 km

Ejemplo 4:

```
novedades1 = [3, "18", 1, "20", 2, "25", 1, "30", 1, "P", 3  
              , "10"]
```

```
historial_carrera = avances(novedades1, [])
```

Resultado:

```
historial_carrera = [1, 50, 2, 25, 3, 28 ]
```