



Wheeljack reportó a Optimus que, para prepararse mejor para las siguientes batallas contra Megatron, lo ideal sería poder medir las distintas habilidades en velocidad y ataque de los distintos Autobots¹ y así poder evaluar cómo mejorar a cada uno. Por lo tanto, mientras él se encarga de reconstruir a los Dinobots² para reforzar al equipo, nos manda los siguientes requerimientos del programa en Haskell que requerirá para realizar las mediciones.

En primer lugar para representar a un Autobot tendremos la siguiente estructura:

```
data Autobot =  
    Robot String (Int,Int,Int) ((Int,Int,Int) -> (Int,Int)) |  
    Vehiculo String (Int,Int)
```

Cuando son Robots constan de un nombre, una tupla de capacidades fuerza, velocidad y resistencia, y una función que calcula sus valores para transformarse en vehículo.

En cambio, cuando se transforman en Vehículos, no tienen fuerza, debido a que les es imposible atacar, pero tanto su velocidad como su resistencia se ven aumentadas de forma particular en cada uno.

```
optimus = Robot "Optimus Prime" (20,20,10) optimusTransformacion  
optimusTransformacion (_,v,r) = (v * 5, r * 2)
```

```
jazz = Robot "Jazz" (8,35,3) jazzTransformacion  
jazzTransformacion (_,v,r) = (v * 6, r * 3)
```

```
wheeljack = Robot "Wheeljack" (11,30,4) wheeljackTransformacion  
wheeljackTransformacion (_,v,r) = (v * 4, r * 3)
```

```
bumblebee = Robot "Bumblebee" (10,33,5) bumblebeeTransformacion  
bumblebeeTransformacion (_,v,r) = (v * 4, r * 2)
```

```
autobots = [ optimus, jazz, wheeljack, bumblebee ]
```

¹ Los Autobots son robots que se transforman en vehículos. Hay excepciones que se transforman en otra cosa, pero no vamos a considerarlas.

² Los Dinobots también son robots, aliados de los Autobots, pero se transforman en dinosaurios.

Importante: Leer muy bien en cada caso si se habla de Autobot, Robot o Vehículo.

Se pide:

1. Definir la función **maximoSegun/3** que recibe una función (que recibe dos valores) y dos valores y devuelve aquel valor que maximiza la función al ser invocada con dicho valor como primer parámetro. En caso de empate, devolver el primero de los valores.

Ejemplo:

```
> maximoSegun (-) 3 4
```

```
4 --> Porque (4 - 3) > (3 - 4) ... es lo mismo que: ((-) 4 3) > ((-) 3 4)
```

2. Implementar las diferentes funciones que permiten acceder a los atributos de los Autobots sin importar si están transformados o no. Como se dijo anteriormente, la fuerza de un vehículo es 0.
3. Crear la función **transformar/1** que permita cambiar un Autobot de un Robot a un Vehículo, teniendo en cuenta que la función de conversión se le aplica a los atributos del Robot.
4. Realizar la función **velocidadContra/2** que permita calcular la velocidad de un Autobot contra otro, teniendo en cuenta que la diferencia entre la fuerza del segundo menos la resistencia del primero se le resta a la velocidad del primero, siempre y cuando dicha diferencia sea positiva. Es decir que si fuerza del segundo es 30 y la resistencia del primero es 40, la diferencia es -10 y no debería afectar al primero.
5. Definir la función **elMasRapido/2** que dados dos Autobot, devuelve aquel que tiene la mayor velocidad, teniendo en cuenta que la velocidad de uno es afectada por el otro como se indica en el punto anterior.

6.

- a. Implementar la función **domina/2** que dados dos Robots indica si el primero tiene igual o mayor velocidad que el segundo *siempre*, es decir en todas sus formas: Robot vs. Robot, Robot vs. Vehículo, Vehículo vs. Robot, Vehículo vs. Vehículo.

Nota: ¡Evitar repetición de código!

Tip: Tuplas para los emparejamientos y el uso de `uncurry` pueden ser muy útiles.

- b. Realizar la función **losDominaATodos/2** que recibe un Robot y una lista de otros Robots e indica si el primero los domina a todos los de la lista.

7.

- a. Definir una función **quienesCumplen/2** que dada una condición y una lista de Autobots, obtenga una lista con los nombres de los Autobots que cumplen la condición.
- b. Usar la función anterior en una consulta para saber si algún Autobot domina a todos los demás y su nombre termina en vocal.

Nota: No usar funciones locales, lambda y/o auxiliares que no estén ya definidas para puntos anteriores.

8. Indicar el tipo de la siguiente función:

```
saraza x y w z = z w . maximoSegun z y $ x
```