

Enterprise' warp-drive management software

La *Starship Enterprise* se dirige al espacio *Klingon* en una misión secreta. Durante una operación rutinaria, Scotty detecta que el ordenador que gestiona el motor *Warp* carece de un sistema de gestión de potencia para los tres inyectores de plasma que regulan la energía del reactor de antimateria, y por tanto la velocidad de la nave.

Conocedor de la fiereza de los *Klingons*, decide implementar una aplicación que le ofrezca el mejor balance entre los tres inyectores teniendo en cuenta que pueden resultar dañados en el combate.

El flujo de plasma al reactor tiene una relación lineal con la velocidad. Para acelerar al 100% de la velocidad de la luz, el reactor de antimateria necesita un flujo de plasma de 300 mg/s (miligramos por segundo). Cada inyector puede inyectar un flujo de 100 mg/s.

Cada inyector puede funcionar al 100% de su capacidad indefinidamente. No obstante, puede funcionar inyectando un flujo de hasta 99 mg/s por encima de su capacidad. Por cada mg/s de flujo por encima de su capacidad, pierde un minuto de funcionamiento, siendo el tiempo de funcionamiento del primer mg/s extra 99 minutos, y el tiempo de funcionamiento con 99 mg/s extras, 1 minuto.

Además, el flujo máximo al que puede funcionar indefinidamente baja en función del daño del inyector: por cada punto de daño, baja un mg/s. Si está dañado al 70%, su flujo máximo indefinido es 30 mg/s. Si está dañado al 32%, el flujo máximo es de 68 mg/s. Un inyector puede seguir inyectando 99 mg/s extras a pesar de estar dañado, así un inyector dañado al 20% puede inyectar 179 mg/s como máximo y funcionar durante 1 minuto. Si un inyector está dañado al 100%, no es utilizable.

Se debe desarrollar un código que permita **calcular el flujo de funcionamiento de cada inyector para un porcentaje de la velocidad de la luz deseado**, de modo que maximice el tiempo de funcionamiento en una situación de daño dada. Las soluciones no deben contemplar el agotar el tiempo de funcionamiento de cada inyector de forma secuencial.

También, para una situación de daño dada, debe ofrecer la velocidad máxima a la que se puede viajar de forma indefinida.

A modo de ejemplo, se ofrecen los siguientes *inputs* con sus respectivos resultados esperados:

Caso	Daño inyector A	Daño inyector B	Daño inyector C	Porcentaje de la velocidad de la luz	Resultado
1	0%	0%	0%	100%	A: 100 mg/s, B: 100 mg/s, C: 100 mg/s
2	0%	0%	0%	90%	A: 90 mg/s, B: 90 mg/s, C: 90 mg/s
3	0%	0%	0%	30%	A: 30 mg/s, B: 30 mg/s, C: 30 mg/s
4	20%	10%	0%	100%	A: 90 mg/s, B: 100 mg/s, C: 110 mg/s
5	0%	0%	100%	80%	A: 120 mg/s, B: 120 mg/s, C: 0 mg/s
6	0%	0%	0%	150%	A: 150 mg/s, B: 150 mg/s, C: 150 mg/s
7	0%	0%	30%	140%	A: 150 mg/s, B: 150 mg/s, C: 120 mg/s
8	20%	50%	40%	170%	"Unable to comply"

A continuación, para cada *input* propuesto más arriba, el código debe ser capaz de calcular el tiempo máximo de funcionamiento. Los resultados esperados son:

Caso	Tiempo de funcionamiento
1	Infinito
2	Infinito
3	Infinito
4	90 minutos
5	80 minutos
6	50 minutos
7	50 minutos
8	0 minutos