

Sanando Luchadores

Contribución de Facundo Gutiérrez y Agustín Santiago Gutiérrez

Descripción del problema

Shining Force es un videojuego de fantasía de tipo RPG táctico, lanzado por primera vez en 1992.

Durante el transcurso del juego ocurren batallas entre las fuerzas del mal y un grupo de valientes luchadores liderados por el jugador. Cada luchador posee al comenzar la batalla una cierta cantidad de **puntos de vida** y de **puntos de magia**. A medida que los luchadores son golpeados por el enemigo pierden puntos de vida, y a medida que utilizan habilidades mágicas consumen sus puntos de magia.

Los luchadores se dividen en diferentes clases de acuerdo a sus habilidades: hay espadachines agresivos, poderosas hechiceras, arqueras eficaces, enanos robustos con grandes hachas, hombres pájaro y muchas clases más de luchadores.

Una de las clases más importantes en el juego es la clase de los **sanadores**. Los sanadores son personajes que tienen la capacidad especial de sanar a otros luchadores, haciendo que **recuperen puntos de vida** perdidos durante el combate. Cada vez que se utiliza esta capacidad especial, se consumen puntos de magia del sanador.

En este problema, tienes a tu disposición un sanador con las siguientes capacidades:

1. **Consumiendo 3 puntos de magia**, se recuperan **hasta un máximo de 10 puntos de vida** de un luchador elegido, **que esté a distancia 1** del sanador
2. **Consumiendo 5 puntos de magia**, se recuperan **hasta un máximo de 10 puntos de vida** de un luchador elegido, **que esté a distancia 1 o 2** del sanador

Nunca está permitido que un luchador alcance más puntos de vida que los que tenía al comenzar la batalla, sin importar cuántas veces sea sanado.

Debes implementar una función que, sabiendo la cantidad de puntos de magia de un sanador y los puntos de vida actuales e iniciales de cada uno de n luchadores, así como su distancia al sanador, determine la máxima cantidad posible de puntos de vida que es posible recuperar en total.

Detalles de implementación

Se debe implementar una función:

`sanar(magia, vidaInicial, vidaActual, distancia)`, donde:

- `magia`: Un entero no negativo, que indica la cantidad total de puntos de magia que tiene disponibles el sanador.
- `vidaInicial, vidaActual, distancia`: Arreglos de n enteros. El luchador i -ésimo tiene actualmente `vidaActual[i]` puntos de vida, tenía `vidaInicial[i]` puntos de vida al comenzar la batalla, y se encuentra a una distancia `distancia[i]` del sanador.

La función debe retornar un único entero: la máxima cantidad de puntos de vida que es posible sanar en total.

Evaluador local

El evaluador local lee de la entrada estándar:

- Una línea con dos enteros n y $magia$
- n líneas, cada una con tres enteros $vidaInicial[i]$, $vidaActual[i]$ y $distancia[i]$

El evaluador escribe a la salida estándar una única línea, con el entero retornado por la función `sanar`.

Restricciones

- $1 \leq n \leq 50$
- $0 \leq magia \leq 100$
- $1 \leq vidaActual[i] \leq vidaInicial[i] \leq 100$
- $1 \leq distancia[i] \leq 2$

Ejemplos

Si el evaluador local recibe la siguiente entrada:

```
3 12
20 6 1
30 22 2
40 27 1
```

Con una solución correcta escribe:

```
28
```

Lo que se obtiene sanando una vez a cada luchador (lo que consume $3 + 5 + 3 = 11$ puntos de magia).

Si en cambio la entrada fuera:

```
3 12
20 1 1
30 22 2
40 27 1
```

La salida sería:

```
32
```

Lo que se obtiene sanando dos veces a cada uno de los luchadores a distancia 1 (lo que consume $3 + 3 + 3 + 3 = 12$ puntos de magia).

Subtareas

1. $n = 1$ (10 ptos.)
2. $n = 2$ (13 ptos.)
3. $magia = 12$ (15 ptos.)
4. $distancia[i] = 1$ para todo i (30 ptos.)
5. Sin más restricción (32 ptos.)