



Amazing Robots

TASK

Tú eres el propietario de dos robots que están situados en dos laberintos rectangulares separados. El cuadrado (1,1) en un laberinto es el cuadrado de la esquina superior izquierda, que es la esquina norte-oeste. El laberinto i ($i = 1, 2$) tiene un conjunto de G_i ($0 \leq G_i \leq 10$) de guardas que intentan capturar a los robots patrullando hacia atrás y hacia adelante sobre un camino recto. El objetivo es determinar una secuencia de comandos tales que los robots salgan del laberinto sin que ninguno de ellos sea capturado por un guarda.

Al principio de cada minuto, tú envías el mismo mensaje a ambos robots. Cada comando es una dirección (norte, sur, este u oeste). Un robot se mueve un cuadrado en la dirección del comando, excepto si el robot colisionará con una pared, en cuyo caso el robot no se mueve en este minuto. Un robot sale del laberinto si camina fuera de él. Un robot ignora los comandos una vez que ya está afuera.

Los guardas se mueven un cuadrado al principio de cada minuto, al mismo tiempo que los robots. Cada guarda empieza en un cuadrado dado, mirando a una dirección dada y avanzando un cuadrado por minuto hasta que el guarda ha hecho un número de movimientos igual al número de cuadrados de la ruta que patrulla menos uno. El guarda entonces gira instantáneamente y camina en la dirección opuesta hasta su cuadrado inicial, donde gira de nuevo y repite su ruta de patrulla hasta que cada robot haya salido de su laberinto.

La ruta de patrulla de un guarda no requiere que el guarda pase a través de paredes o que salga del laberinto. Sin embargo, las rutas de patrulla pueden solaparse, dos guardas nunca colisionarán: nunca ocuparán el mismo cuadrado al final de un minuto, y ellos nunca intercambiarán sus cuadrados durante un minuto. Un guarda en un laberinto no empezará en el mismo cuadrado que un robot.

Un guarda captura un robot si el guarda ocupa el mismo cuadrado que el robot al final de un minuto, o si el guarda y el robot intercambian sus cuadrados durante un minuto.

Dados dos laberintos (cada uno no mayor de 20x20) con el cuadrado inicial para cada robot y con las rutas de patrulla de los guardas en cada laberinto, determina la secuencia de comandos para los cuales los robots salen sin ser capturados por los guardas. Minimiza el tiempo que tardará para sacar al último robot de su laberinto. Si los robots salen en diferentes tiempos, el tiempo en que sale el primer robot no tiene importancia.

Entrada: `robots.in`

El primer conjunto de líneas describen el laberinto y sus ocupantes. A continuación, el segundo conjunto de líneas describen al segundo laberinto y a sus ocupantes.

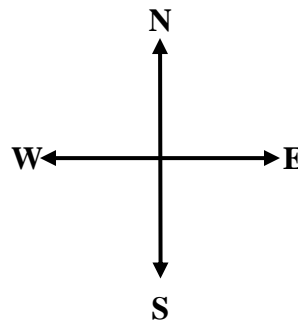
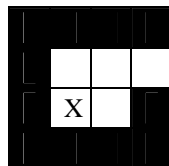
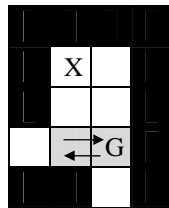


- La primera línea de la entrada contiene dos enteros R_1 y C_1 separados por un espacio, el número de filas y columnas del laberinto 1.
- Las siguientes R_1 líneas cada una contiene C_1 caracteres especificando la apariencia del laberinto. El cuadrado inicial del robot está especificado con una 'X'. Un '.' representa un espacio vacío y '#' representa una pared. Cada laberinto contiene exactamente un robot.
- Siguiendo la apariencia del laberinto, hay una sola línea con un número entero G_1 , el número de guardas en el primer laberinto ($0 \leq G_1 \leq 10$).
- Cada una de las siguientes G_1 líneas describen un guarda con 3 enteros y un carácter, separados por espacio. Los dos primeros enteros describen la fila y la columna del cuadrado inicial del guarda. El tercer entero especifica el número de cuadrados (2...4) de la ruta del guarda. El carácter especifica la dirección inicial del guarda: 'N', 'S', 'E' o 'W' (norte, sur, este y oeste, respectivamente).

La descripción del segundo laberinto sigue a la descripción del primero, en el mismo formato, pero potencialmente con valores diferentes.

Ejemplo de entrada:

```
5 4
####
#X.#
#...#
...#
##.#
1
4 3 2 W
4 4
####
#...
#X.#
####
0
```



Salida: robots.out

La primera línea de la salida debe contener un entero K ($K \leq 10000$), el número de comandos para que ambos robots salgan del laberinto sin ser capturados. Si tal secuencia de comandos existe, la menor secuencia no tendrá más de 10000 comandos. Las siguientes K líneas son la secuencia de comandos, cada una contiene un solo carácter del conjunto {'N', 'S', 'E', 'W'}. Si tal secuencia no existe, saca una sola línea conteniendo "-1".

Ambos robots deben salir de sus laberintos al acabar los comandos. El último comando debe provocar que al menos uno de los robots salga del laberinto. Si diversas secuencias de comandos provocan que los robots salgan en el mínimo tiempo, cualquiera de ellas será aceptada.



Ejemplo de salida:

8
E
N
E
S
S
S
E
S

RESTRICCIONES

Running time	2 seconds of CPU
Memory	64 MB

PUNTUACIÓN

No se dará crédito parcial para los test en los que no exista una secuencia de comandos. Se da puntuación parcial para otros casos tal como se describe a continuación.

Corrección: 20% de los puntos

El fichero de salida para un test se considera correcto si tiene el formato correcto, no tiene más de 10000 comandos y la secuencia de comandos provoca que los robots salgan del laberinto, con el último comando que provoca que al menos un robot salga de su laberinto.

Minimalidad: 80% de los puntos

El fichero de salida se considera minimal si es correcto y no hay otra secuencia de comandos más corta que sea correcta. Un programa cuya secuencia de comandos no es minimal no recibirá puntos por minimalidad.