

Preparando pociones

Contribución de Agustín Santiago Gutiérrez

Descripción del problema

Herminia es una joven alquimista aprendiz, y tiene N pociones en una hilera. Cada poción es de uno de 10 tipos posibles, que identificamos con letras desde a hasta j en orden alfabético.

En un paso, Herminia elige dos pociones consecutivas de la hilera (que sean combinables) y las combina, reemplazando ambas por la poción resultante, en la misma posición de la hilera en que estaban ambas pociones consecutivas. Si realiza $N-1$ pasos de este tipo, Herminia logrará obtener una única poción como resultado final de toda la combinación.

Existe una tablita de 10×10 en el famoso libro "Introduction to Alchemy" que indica, para cada posible par de tipos de pociones, si ese par de pociones se puede combinar en ese orden (ya que quizás, no es posible combinar "c a la izquierda y f a la derecha", pero sí "f a la izquierda y c a la derecha"). Además si es posible esa combinación, la tablita indica cuál será el tipo de poción resultante para ese par en ese orden (siempre en a-j).

Por ejemplo podría ser que ad no se pueda combinar, pero que ac sí y el resultado de la combinación sea f. También podrían pasar cosas como que ee no fuera combinable, pero ff sí, y el resultado de combinar dos pociones f sea una poción c. La alquimia es un arte muy compleja.

Dos maneras de combinar todas las pociones en una única poción final se consideran la misma, si realizan exactamente las mismas combinaciones de pociones, aunque puedan hacerlo en otro orden.

Por ejemplo si inicialmente la hilera tiene 4 pociones abcd, combinar en el primer paso ab y en el segundo cd, para

finalmente en el tercer paso combinar las dos pociones resultantes entre sí, se considera exactamente la misma forma de combinar que si en el primer paso se hubiera elegido cd, y en el segundo ab, para nuevamente en el tercer paso combinar las dos pociones restantes.

Si en cambio se hubiera combinado ab en el primer paso, pero en el segundo paso se combinara la poción obtenida en el primer paso con la c, para luego finalmente combinar el resultado con la d en un último paso, esto se considera una forma de combinar diferente de la anterior.

Otro ejemplo podría ser tres pociones de tipo a: suponiendo que combinar dos pociones a produce una poción a, entonces existen en este caso dos maneras distintas de combinar todo, dependiendo del par de pociones tipo a consecutivas que elijamos combinar primero.

La manera de combinar es crítica tanto para el resultado final como para los resultados intermedios: por razones que los grandes alquimistas conocen, pero que escapan aún a la comprensión de Herminia, no todas las distintas maneras de combinar las mismas pociones dan lo mismo, incluso si se llega a una poción del mismo tipo como resultado. Algunas formas de combinar pueden ser más difíciles y peligrosas para el alquimista que las practica. Esto es un problema para Herminia porque aún no entiende bien qué maneras de combinar son peligrosas, y qué maneras son seguras.

Debido a esto, la hilera de N pociones se dice *peligrosa*, si existe en ella alguna subcadena de 2 o más pociones consecutivas (la subcadena podría ser incluso toda la hilera), tal que es posible combinar todas esas pociones de la subcadena entre sí (sin usar otras que hubiera en la

hilera) mediante pasos permitidos, de al menos 2 maneras diferentes.

Dada la tablita de 10x10 sacada del Introduction to Alchemy, y la cadena de **N** pociones (letras) inicialmente en hilera, tu programa debe ayudar a Herminia a determinar si la hilera es peligrosa. En caso de que la hilera no sea peligrosa, se pide indicar además cómo combinar las **N** pociones de la hilera para obtener una única poción final, en caso de que esto sea posible.

Descripción de la función

Debes implementar la función `pociones(tablita, hilera, combinaciones)`, que recibe:

- `tablita`: Un arreglo de 10 cadenas, de 10 caracteres cada una. `tablita[i][j]` indica la letra resultante de combinar una poción del *i*-ésimo tipo a la izquierda, con una del *j*-ésimo tipo a la derecha. Si ese par en ese orden no es combinable, `tablita[i][j]` tendrá un carácter "."
- `hilera`: Una cadena de **N** letras entre **a** y **j** inclusive, que indican los tipos de las pociones en el orden de la hilera.
- `combinaciones`: Arreglo en el que escribir una lista de enteros, indicando las **N - 1** combinaciones de pociones que se deben hacer para obtener una única poción final, en el orden en que se realizan. El entero **x** indica que en ese paso, se deben combinar las pociones en las posiciones **x** y **x + 1** de la hilera, contando desde **1**. Es decir, **1** indica que se deben combinar las dos primeras pociones de la hilera (de acuerdo al estado de la hilera en ese paso), **2** indica que se deben combinar la segunda y la tercera, y así siguiendo.
 - Si la hilera es peligrosa, se debe

dejar vacío.

- Si la hilera no es peligrosa pero no existe forma de combinar todas las pociones para llegar a una sola, también se debe dejar vacío.
- Si hubiera más de una solución válida, se debe dar aquella que tenga un número lo más chico posible en `combinaciones[0]` (que corresponde al primer paso). Si sigue habiendo empate, entre ellas aquella con un número lo más chico posible en `combinaciones[1]` y así siguiendo.

La función debe retornar la cadena de texto "PELIGROSA" si se determina que la hilera de pociones es peligrosa, o "SEGURA" en caso contrario.

Evaluador

El evaluador local lee de la entrada estándar:

- Una línea con la cadena `hilera`
- **10** líneas más, cada una con una cadena de **10** caracteres. Estas líneas indican los valores del arreglo `tablita`.

Escribe a la salida estándar una primera línea el valor retornado por la función, y una segunda línea con el valor almacenado en el parámetro `combinaciones`.

Restricciones

- $2 \leq N \leq 5.000$
- Todas las letras de las cadenas que recibe la función estarán en minúscula, desde **a** hasta **j** inclusive (o "." en el caso de `tablita`).

Ejemplo

Si el evaluador recibiera:

```
baaa  
c.c.....  
j.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
.....  
..d.....
```

Para un programa correcto devuelve:

```
SEGURA  
1 2 1
```

Que corresponde a combinar **baaa** → **jaa** → **jc** → **d**

Si en cambio recibiera:

```
baaa  
d..d.....  
c.....  
e.....  
d.....  
e.....  
.....  
.....  
.....  
.....  
.....  
abcdefghij
```

Devolvería:

```
PELIGROSA
```

Es peligrosa porque la subcadena **aaa** se puede combinar haciendo **aaa** → **da** → **d** o haciendo **aaa** → **ad** → **d**. Notar que en

este caso, existe una única forma posible de combinar toda la hilera, pero aún así la consideramos peligrosa por culpa de la subcadena **aaa**.

Puntuación

Se obtiene el 40 % del puntaje por el valor de retorno correcto, y el 60 % restante por además dar el resultado correcto en combinaciones.

Subtareas

1. `tablita[i][j] == '.'` a menos que sea $i = 0$ y $j > 0$ (10 puntos)
2. `tablita[i][j] == '.'` a menos que sea $i = 0$ (10 puntos)
3. $N \leq 10$ (5 puntos)
4. $N \leq 200$ (25 puntos)
5. $N \leq 2000$ (35 puntos)
6. Sin más restricción (15 puntos)