

## Produciendo Galletitas

Contribución de Iván Sadofski Costa

### Descripción del problema

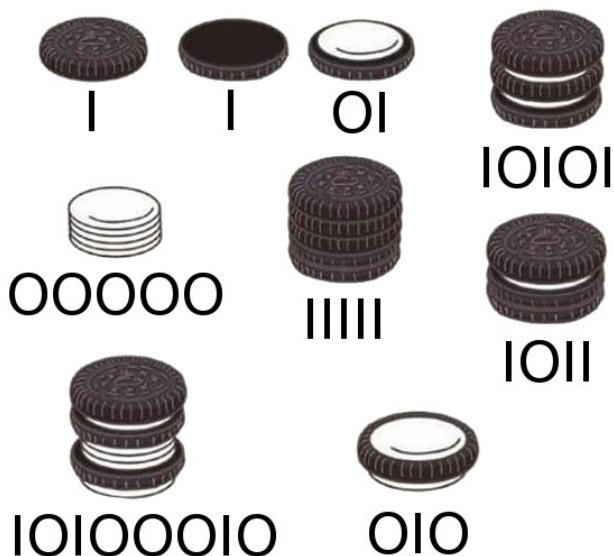
Una reconocida fábrica de galletitas desea auspiciar la IOI. Han decidido que la mejor forma de hacerlo es adaptar una de sus galletitas más exitosas para crear nuevas galletitas temáticas de IOI.

La galletita IOI básica tiene el siguiente formato, que consiste en dos tapas de chocolate encerrando una capa de relleno:



IOI

La organización de la IOI desea aumentar la variedad de galletitas posibles, por lo cual solicita a la fábrica diferentes modelos especiales. Cada modelo posible de galletita se identifica por medio de una palabra de  $N$  letras, en la cual cada letra I indica una tapa de chocolate, y cada letra O indica una capa de relleno, en orden, de arriba hacia abajo.



La orientación en el espacio de una tapa de chocolate, o de una capa de relleno, no se consideran importantes: por ejemplo, todas las galletitas OIII y todas las IIIO se consideran indistinguibles, ya que tienen el mismo diseño. Lo mismo ocurre con OI e IO.

El proceso de producción de una galletita de uno de estos nuevos diseños utiliza como materia prima las galletitas IOI originales, ya que su producción fue optimizada durante años en enormes fábricas, y por lo tanto son extremadamente baratas.

Para producir una galletita, se comienza seleccionando una cierta cantidad  $K$  de galletitas IOI, que serán toda la materia prima que se utilizará para producir la nueva galletita. Luego, se pueden realizar las siguientes operaciones, en cualquier orden y tantas veces como sea necesario:

1. **Partir una de las  $K$  galletitas IOI originales en dos partes:** una IO (que equivale a una OI) y una I.
2. **Partir una de las  $K$  galletitas IOI originales en tres partes:** dos I y una O.
3. **Unir dos galletitas (o partes de galletitas) existentes.** Las dos partes se unen colocando una encima de la otra, en cualquier orden que se desee.

Así por ejemplo, se podría formar la IOIOI a partir de dos galletitas IOI, si primero se parte una de ellas en dos, teniendo así IOI, IO e I, y finalmente uniendo IOI con IO se obtiene IOIOI. En este caso, quedaría una tapa I sobrante, lo cual está permitido.

Cada galletita de materia prima utilizada tiene un costo **G**. Cada operación de tipo 1 tiene un costo **D** (partir una galletita IOI en dos). Cada operación de tipo 2 tiene un costo **T** (partir una galletita IOI en tres). Las operaciones de tipo 3 no tienen costo alguno (unir dos galletitas). **G, D, T** son enteros no negativos, que expresan los costos en una misma unidad.

Debes escribir una función que dado el diseño de galletita deseado, y los costos **G, D, T**, retorne el mínimo costo de producción posible, y además calcule un plan de producción óptimo, es decir, uno que produzca las galletitas con el mínimo costo posible.

### Descripción de la función

Debes implementar la función `galletitas(cadena, G, D, T)`, que recibe:

- `cadena`: Una cadena de **N** caracteres que describe el diseño deseado. Contendrá únicamente caracteres **I** y **O**.
- `G, D, T`: Enteros que indican los costos de las respectivas operaciones.

La función retorna un entero con el mínimo costo de producción posible.

Además, la implementación de `galletitas` puede llamar a la función `parte(tam, i)`, indicando respectivamente el tamaño `tam` de la siguiente porción de la galletita final que se colocará, y el índice `i` de la galletita inicial de donde proviene esa porción:  $1 \leq \text{tam} \leq 3$ ,  $1 \leq i \leq K \leq N$ .

Para el ejemplo anteriormente explicado del diseño IOIOI, se podría llamar primero a `parte(3, 1)`, para indicar que los primeros 3 caracteres (IOI) provienen de la galletita 1. Y luego se llamaría a `parte(2, 2)` para indicar que **los siguientes 2** caracteres (que ahora serán OI) provienen de la galletita 2. Notar que estas partes deben irse indicando **en el orden de la cadena de entrada**.

Otro plan posible sería llamar primero `parte(2, 2)`, luego a `parte(1, 1)` y luego a `parte(2, 1)`. Este plan corresponde a utilizar dos galletitas y cortar ambas en 2, utilizando la parte IO de la galletita 2, y las partes I y OI de la galletita 1.

### Evaluador

El evaluador local lee de la entrada estándar con el siguiente formato:

- En la primera línea, una cadena de texto con el diseño de galletita requerido.
- En la segunda línea, enteros **G, D, T** con los costos correspondientes.

El evaluador local escribe a la salida estándar una primera línea con el entero retornado por la función `galletitas`. Luego imprime una segunda línea con la cantidad total de llamados realizados a la función `parte`. Finalmente, imprime una línea adicional por cada llamado a esta función, indicando los parámetros `tam, i` utilizados.

### Cotas

- $1 \leq N \leq 200.000$
- $0 \leq G, D, T \leq 1.000$
- $D \leq T$

**Ejemplo**

Si se invoca al evaluador con la siguiente entrada:

IOIOI
10 1 2

Para un programa correcto, la salida podría ser:

21
2
3 1
2 2

También podría ser:

21
2
3 2
2 1

Y también podría ser:

21
2
2 2
3 1

**Puntuación**

Se obtiene el 40% del puntaje por calcular correctamente el costo óptimo, y el 60% restante por además calcular un plan de producción óptimo.

**Subtareas**

- $G = D = T = 0$ : 5 puntos
- $G = 1, D = 0, T = 2$ : 10 puntos
- $G = 2, D = 0, T = 1$ : 10 puntos
- $G = D = T = 1$ : 10 puntos
- $N \leq 15$ : 10 puntos
- $N \leq 100$ : 20 puntos
- Sin más restricciones: 35 puntos