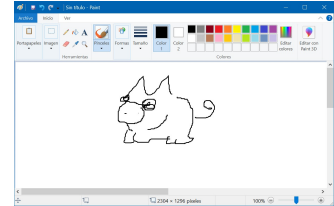


## Compresión RLE



Muchos de los que hemos utilizado Windows no hemos podido resistir la tentación de abrir *Paint* alguna vez y pintarrapear garabatos en el lienzo. Este programa, que viene preinstalado con Windows desde su versión 1.0, ya es un dinosaurio de la informática, aunque no siempre se ha llamado *Paint*. En versiones de Windows anteriores a Windows 95 se llamaba *Paintbrush*, y guardaba las imágenes en formatos BMP y PCX, que fueron muy populares en aquella época, pero que hoy están en desuso. Ambos formatos utilizaban un mecanismo de compresión muy primitivo: *Run-length encoding*, o *RLE*. Este mecanismo funciona del siguiente modo: supongamos que tenemos una imagen de 10 píxeles de ancho y 3 píxeles de alto, y dos colores (blanco y negro).



Si no utilizáramos ningún mecanismo de compresión, podríamos representar esta imagen mediante una secuencia de píxeles, comenzando por la primera fila (de izquierda a derecha), luego por la segunda, etc. De este modo, tendríamos la siguiente secuencia, donde N representa un píxel de color negro, y B representa uno de color blanco:

N N N N N N N N N N B B B N B B B N N N N N N N N

La compresión RLE se beneficia del hecho de que esta secuencia suele contener varios píxeles seguidos del mismo color, formando un *tramo*. Por ejemplo, en la secuencia anterior, tenemos un tramo de 11 píxeles de color negro, seguido de otro tramo de 3 píxeles de color blanco, y así sucesivamente. De este modo, podemos representar nuestra imagen mediante una secuencia de tramos. Para cada tramo indicamos su longitud y el color de todos sus píxeles. De este modo, nuestra imagen se representaría mediante la siguiente secuencia:

11 N 4 B 1 N 3 B 11 N

He conseguido recuperar de un ordenador antiguo unos dibujos que hice en *Paintbrush* cuando era pequeño. Por desgracia, están en formato PCX y no tengo ninguna aplicación moderna que los muestre. ¿Puedes ayudarme a averiguar su contenido?

*Indicación.* Utiliza la clase `map` de la STL de C++ para resolver este ejercicio.

## Entrada

La entrada contiene una serie de casos de prueba. Cada uno de ellos se compone de:

- Una línea con dos números  $W$  y  $H$  que indican el ancho y alto (en píxeles) de la figura, respectivamente ( $1 \leq W \leq 2 \cdot 10^4$ ,  $1 \leq H \leq 2 \cdot 10^4$ ).

- Una línea con el número  $N$  de tramos de los que se compone la imagen ( $1 \leq N \leq 10^6$ ). A continuación, otra línea con la descripción de los tramos. Para cada tramo se indica su longitud y el color de sus píxeles. Cada color se representa mediante una cadena de, a lo sumo, 10 caracteres alfanuméricos sin espacios.
- Una línea con el número  $Q$  de consultas que quiero realizar sobre la imagen ( $1 \leq Q \leq 10^5$ ), seguida de otras  $Q$  líneas, cada una de ellas con las coordenadas  $(X, Y)$  del punto que quiero consultar ( $1 \leq X \leq W, 1 \leq Y \leq H$ ). Las coordenadas  $(1, 1)$  corresponden a la esquina superior izquierda de la imagen.

La entrada finaliza con dos ceros, caso que no se procesa.

## Salida

Para cada caso de prueba deben imprimirse  $Q$  líneas, cada una de ellas indicando el color del píxel correspondiente a las coordenadas de cada consulta. Al final de cada caso de prueba debe imprimirse una línea con tres guiones (---).

### Entrada de ejemplo

```
10 3
5
11 N 4 B 1 N 3 B 11 N
3
1 1
5 2
6 2
4 4
5
5 Rojo 2 Verde 2 Rojo 2 Verde 5 Rojo
2
2 2
4 4
0 0
```

### Salida de ejemplo

```
N
B
N
---
Verde
Rojo
---
```

## **Créditos**

**Autor:** Manuel Montenegro.

**Revisores:** Marco Antonio Gómez, Pedro Pablo Gómez, Isabel Pita, Rubén Rubio y Alberto Verdejo.