

Espacios blancos en la imagen

Tenemos una imagen representada por una matriz de colores. Cada color se representa según el código RGB con representación hexadecimal. El código RGB representa un color mediante la mezcla de los tres colores primarios (Red, Green y Blue). En la representación hexadecimal, la intensidad de cada color en la mezcla se indica mediante dos dígitos hexadecimales. Así el negro (ausencia de todo color) tiene la representación 000000, mientras que el blanco tiene la intensidad máxima en todos los colores, FFFFFFFF. Los colores primarios se representan como: el rojo FF0000, el verde 00FF00 y el azul 0000FF. A partir de aquí se pueden combinar con mayor o menor intensidad, por ejemplo, el amarillo: FFFF00, o el gris oscuro 5E5E5E.

Estamos analizando una imagen y queremos encontrar el mayor espacio blanco en cada fila de la imagen. En caso de que haya dos espacios blancos iguales optaremos por el de más a la izquierda.

Requisitos de implementación.

Cada fila de la imagen sólo puede recorrerse una vez.

Debe utilizarse un solo vector y no puede utilizarse una matriz

Se pide especificar la función que debe realizar los cálculos.

Entrada

Cada caso de prueba consta de $m+1$ líneas. En la primera se indica el número m de líneas que tiene la imagen y el número n de pixels de cada línea. Las m líneas siguientes contienen n códigos en hexadecimal que indican el color de cada pixel de la línea.

El tamaño de las imágenes no está acotado.

Salida

Para cada caso de prueba se escriben m líneas, en cada una se indica la posición del primer pixel blanco de la secuencia más larga de blancos y la posición del último blanco de la secuencia. Si no hay ningún pixel en blanco las posiciones serán 0 -1. Los casos se separan por una línea en blanco. Después del último caso no debe dejarse la línea en blanco de la separación.

Entrada de ejemplo

```
5 6
008000 FFFFFFFF FFFFFFFF 0000FF FFFFFFFF 000000
FFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FF0000 FF0000
800080 FFFFFFFF FFFFFFFF 800080 FFFFFFFF FFFFFFFF
FFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
000000 000000 000000 000000 000000 000000
1 7
000000 FFFFFFFF 000000 FFFFFFFF 000000 FFFFFFFF 000000
3 2
808080 808080
FFFFFF FFFFFFFF
000000 FFFFFFFF
```

Salida de ejemplo

```
1 2
0 3
1 2
0 5
0 -1

1 1

0 -1
0 1
1 1
```

Autor: Isabel Pita.