

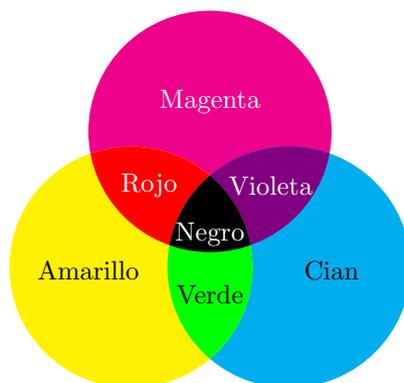
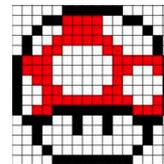
# Pixel Art

Tiempo máximo: 1,000 s Memoria máxima: 4096 KiB

<http://www.aceptaelreto.com/problem/statement.php?id=390>

*Pixel Art* es una técnica que consiste en crear imágenes a partir de pequeños cuadrados o *píxeles* que se rellenan de un único color. Se hizo muy popular en la década de los 80 debido a las limitaciones que tenían los ordenadores para representar imágenes, pero hoy en día sigue siendo popular entre los amantes de lo *retro*.

Reproducir este tipo de imágenes es muy fácil cuando se dispone de todos los colores necesarios. El reto es hacerlo usando sólo los colores primarios: magenta, amarillo y cian. Afortunadamente, la teoría del color nos dice que el resto de colores se puede obtener mezclando otros diferentes según el siguiente esquema:



Por ejemplo, el color rojo se obtiene mezclando magenta y amarillo; el negro, mezclando los 3 colores básicos; y el blanco, dejando el papel sin pintar.

## Entrada

El programa deberá leer, por la entrada estándar, el número de casos de prueba que vendrán a continuación, cada uno en una línea.

Para cada caso, los primeros 3 números indican la cantidad de pintura disponible de cada color básico en este orden: magenta, amarillo y cian. Cada unidad de pintura permite pintar un píxel de ese color. Ten en cuenta que al mezclar colores se gastan unidades de todos los colores usados. Por ejemplo, pintar un píxel de rojo requiere gastar una unidad magenta y otra amarilla.

A continuación aparece la secuencia de píxeles que componen la imagen codificados con letras: magenta (M), amarillo (A), cian (C), rojo (R), negro (N), verde (V), violeta (L) y blanco (B). Todas las imágenes tendrán al menos 1 píxel, y ninguna tendrá más de 100.000.

## Salida

Para cada caso de prueba el programa escribirá una línea. Si la imagen se puede completar con las pinturas disponibles, se escribirá SI y las unidades restantes de pintura magenta, amarillo y cian, en ese orden. Si la imagen no se puede completar, se escribirá NO.

## Entrada de ejemplo

```
3
1 1 1 MCA
3 2 1 MABBCR
1 1 1 RL
```

## Salida de ejemplo

SI 0 0 0
SI 1 0 0
NO

**Autor:** Antonio Sánchez Ruiz-Granados.

**Revisores:** Alberto Verdejo, Marco Antonio Gómez Martín y Pedro Pablo Gómez Martín.