

## Problema D

### Jaque mate

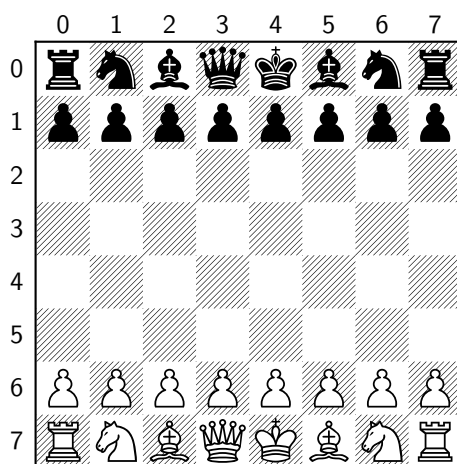
*nombre clave:* jaque

Elizabeth es una maestra mundial de ajedrez. Tras muchas victorias, Elizabeth ha notado que a algunos jugadores les toma demasiado tiempo darse cuenta de que perdieron, por lo que ella debe explicarles el jaque mate.

Tu tarea en este problema es simple. Debes ayudar a Elizabeth creando un programa que, dada la descripción de un tablero, determine si ella tiene a su oponente en jaque mate. A continuación se describen las reglas del ajedrez y cuándo un jugador está en jaque mate.

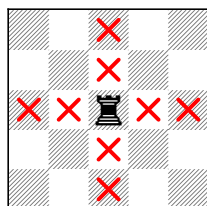
### Reglas del juego

El ajedrez se juega entre dos jugadores sobre un tablero de  $8 \times 8$  casillas. En el contexto de este problema, enumeraremos las columnas del tablero de izquierda a derecha entre 0 y 7, y las filas de arriba hacia abajo entre 0 y 7. Uno de los jugadores controla 16 piezas blancas y el otro controla 16 piezas negras. **Elizabeth siempre controlará las piezas negras.** Existen 6 tipos de piezas: la torre (♖), el alfil (♘), la reina (♕), el rey (♔), el caballo (♞) y el peón (♙). Cada jugador comienza el juego con 8 peones, 2 torres, 2 caballos, 2 alfiles, 1 reina y 1 rey. Al inicio del juego las piezas negras son ubicadas en la parte superior del tablero y las blancas en la parte inferior como se muestra en la siguiente figura.



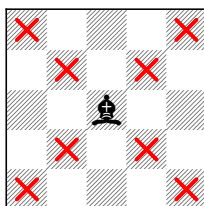
Los jugadores toman turnos y en cada turno deben mover una de sus piezas desde una casilla a otra. Cada tipo de pieza tiene distintas reglas para poder moverse. A continuación se describen las reglas para mover las piezas, excluyendo al peón que será descrito más adelante. Adicionalmente, para cada pieza se muestra una figura marcando con una cruz las posibles casillas en un tablero de  $5 \times 5$  a las que podría moverse cada pieza en un solo movimiento.

### Torre



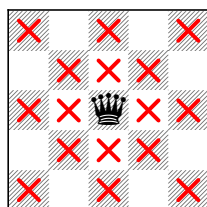
La torre puede moverse en línea recta de forma horizontal o vertical cuantas casillas quiera en un movimiento, pero solo en una dirección.

### Alfil



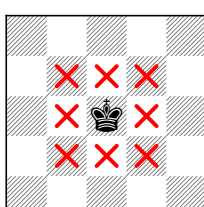
El alfil es similar a la torre, pero en vez de moverse de forma horizontal o vertical, puede moverse a lo largo de una diagonal cuantas casillas quiera en un movimiento.

### Reina



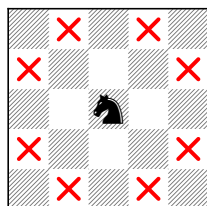
La reina es una combinación de la torre y el alfil ya que puede moverse en diagonal, horizontal o verticalmente cuantas casillas quiera en un movimiento.

### Rey



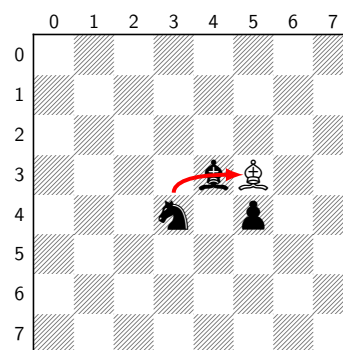
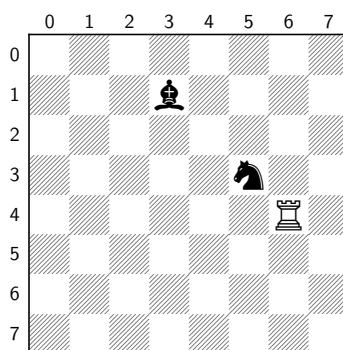
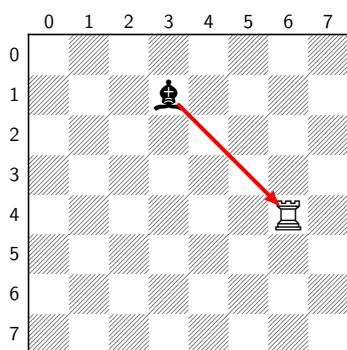
El rey, al igual que la reina, puede moverse en cualquier dirección, pero solo una casilla por movimiento.

### Caballo

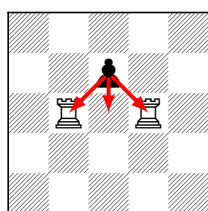


El caballo tiene un movimiento en forma de "L", es decir, siempre se mueve 2 casillas en una dirección (horizontal o vertical), y luego una casilla en la otra dirección.

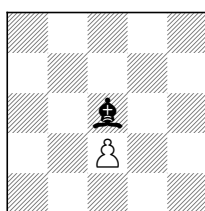
Siguiendo las reglas anteriores, las piezas pueden moverse libremente a cualquier casilla vacía pero sin pasar por encima de otra pieza, a excepción del caballo que puede saltar sobre otras piezas en su movimiento de "L". Adicionalmente, una pieza puede moverse a una casilla que contenga una pieza adversaria, en cuyo caso la pieza adversaria será *capturada* y removida del tablero. Si una pieza  $p$  puede capturar en un movimiento a la pieza  $q$ , diremos que la pieza  $p$  está *atacando* a la pieza  $q$ . A continuación mostramos algunos ejemplos de esto. En el primer ejemplo, el alfil en la casilla (3, 1) está atacando a la torre en la casilla (6, 4). En el segundo, el alfil no está atacando a la torre, pues el caballo en la casilla (5, 3) bloquea el movimiento. Finalmente, en el tercer ejemplo el caballo en la casilla (3, 4) está atacando al alfil en la casilla (5, 3), pues este puede saltar sobre otras piezas.



El peón tiene las reglas de movimientos más complejas del ajedrez. En primer lugar, a diferencia de las otras piezas que pueden avanzar o retroceder, el peón solo puede moverse hacia adelante; es decir, los peones negros solo pueden moverse hacia una casilla cuya fila tenga un índice mayor, mientras que los blancos solo pueden moverse a una casilla cuya fila tenga un índice menor. En segundo lugar, los peones se mueven de forma distinta dependiendo de si se mueven hacia una casilla vacía o si están capturando. Específicamente, un peón puede moverse una casilla hacia adelante si dicha casilla está vacía o puede capturar una pieza adversaria si la pieza está en cualquiera de las casillas diagonales en frente del peón.

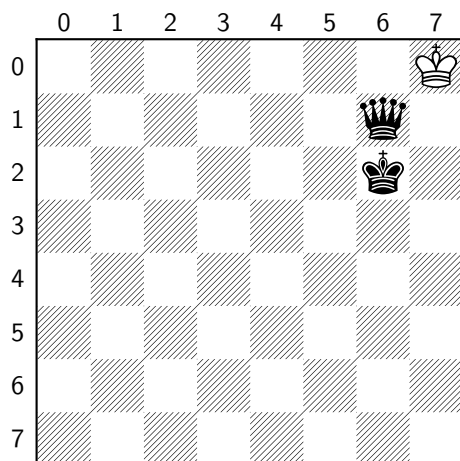


El peón negro puede moverse a la casilla en frente, pues está vacía. También puede moverse a alguna de las diagonales y capturar una de las torres.



No hay movimientos posibles para el peón, pues la casilla en frente está ocupada por el alfil negro y no hay piezas para capturar en las diagonales.

Finalmente, diremos que el rey está en *jaque* si hay una pieza atacándolo. Cuando un jugador tiene su rey en jaque, está obligado a sacarlo del jaque en su siguiente movimiento. Si no existe ningún movimiento con el cual es posible evadir el jaque, se dirá que la *posición* es un *jaque mate* y el jugador cuyo rey está en jaque perderá el juego. La siguiente figura muestra un tablero con una posición que es jaque mate. En primer lugar, el rey blanco está en jaque, pues la reina lo está atacando. Si el rey negro se mueve a las casillas (6,0) o (7, 1), seguirá siendo atacado por la reina y podrá ser capturado en el siguiente turno. Si se mueve a la casilla (6, 1) y captura a la reina, seguirá estando en jaque, pues el rey negro puede capturarlo moviéndose a esta casilla en el siguiente turno. Como no existen más movimientos posibles para el rey blanco, esta posición es jaque mate.



Dada la descripción del tablero, tu tarea es escribir un programa que determine si Elizabeth (piezas negras) tiene a su oponente en jaque mate (piezas blancas).

**Nota** En ajedrez, los peones también tienen reglas distintas cuando se mueven por primera vez. Además existe el *enroque*, la *coronación* y capturar *en passant*. Ninguna de estas reglas debe ser considerada en el contexto de este problema.

## Entrada

La entrada comienza con una línea que contiene un entero  $n$  ( $3 \leq n \leq 32$ ) que representa la cantidad de piezas en el tablero. Las siguientes  $n$  líneas representan cada una de las piezas en el tablero. Cada una de estas líneas contiene 4 enteros  $c, p, y$  y  $x$ .

El entero  $c$  representa el color de la pieza y será **0 si la pieza es blanca** o **1 si la pieza es negra**.

El segundo entero  $p$  es el tipo de la pieza. Este será **0 si es una torre**, **1 si es alfil**, **2 si es reina**, **3 si es rey**, **4 si es caballo** o **5 si es peón**.

El valor  $y$  ( $0 \leq y \leq 7$ ) corresponde al índice de la fila en la cual se encuentra ubicada la pieza. Finalmente, el valor  $x$  ( $0 \leq x \leq 7$ ) corresponde al índice de la columna en que la pieza se encuentra ubicada.

La cantidad de piezas respetará las reglas descritas para el ajedrez. Es decir, la cantidad total de peones de cada color será menor o igual que 8, la de torres menor o igual que 2, la de caballos menor o igual que 2, la de alfiles menor o igual que 2 y la de reinas menor o igual que 1. Adicionalmente, habrá exactamente un rey negro y exactamente un rey blanco. Finalmente, el rey blanco siempre estará en jaque. Es decir, al menos una pieza de color negro estará atacando al rey blanco.

## Salida

Tu programa debe imprimir una única línea con solo un 1 si Elizabeth (piezas negras) tiene a su oponente en jaque mate, o 0 en caso contrario.

## Subtareas y puntaje

### Subtarea 1 (20 puntos)

Se probará varios casos en los que Elizabeth (piezas negras) solo tiene caballos, peones y al rey; y el oponente (piezas blancas) tiene solo una pieza: el rey.

### Subtarea 2 (30 puntos)

Se probará varios casos en que el jaque es múltiple, es decir, hay más de una pieza blanca que está atacando al rey blanco. Puede haber tanto piezas blancas como negras de cualquier tipo.

### Subtarea 3 (50 puntos)

Se probará varios casos sin restricciones adicionales.

## Ejemplos de entrada y salida

### Entrada de ejemplo

3  
0 3 0 7  
1 2 1 6  
1 3 2 6

### Salida de ejemplo

1

### Entrada de ejemplo

5  
0 3 0 0  
0 0 1 5  
1 3 5 4  
1 0 7 0  
1 0 7 1

### Salida de ejemplo

0