

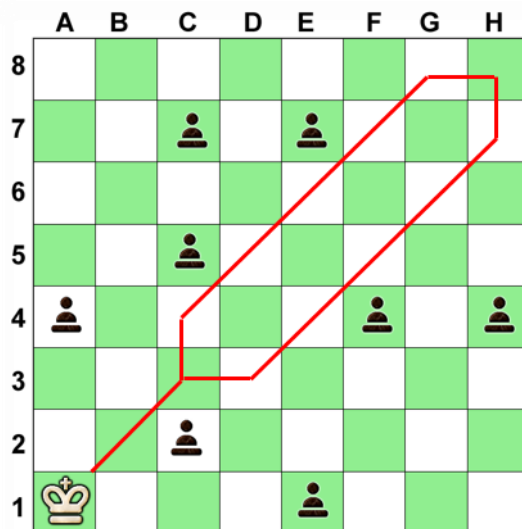
## Camino Seguro del Rey

El rey blanco y 8 peones negros se encuentran ubicados en un tablero de ajedrez (8 casillas de largo por 8 casillas de ancho). El rey y cada uno de los peones pueden "comerse" entre sí. El rey puede moverse de acuerdo a las reglas usuales del ajedrez, esto es una casilla a la vez, en cualquier dirección. Sin embargo, los peones solo pueden moverse para "comer" al rey, para ello el rey tendría que ubicarse en alguna de las casillas diagonales adyacentes a la que ocupa el peón. Diremos que una "ruta" para el rey es SEGURA si el rey puede recorrerla sin ser "comido" por ninguno de los 8 peones. Se trata entonces de encontrar un algoritmo que determine la longitud de la ruta segura MAS CORTA para el rey desde la casilla A1 hasta la casilla H8 para una posición dada de los 8 peones negros. Note que a diferencia de las reglas del ajedrez, en este caso los peones pueden comer en cualquier dirección, siempre que sea una casilla adyacente y diagonal a la posición del peón.

Implemente el método.

```
public static int LongitudMinimaRutaSegura(bool[,] tablero)
```

El parámetro `tablero` representa la distribución de los peones, `true` significa que hay un peón y `false` que no lo hay. El método debe devolver, si existe, la longitud de la ruta segura más corta desde A1 hasta H8. Si no existe tal ruta debe devolver `0`.



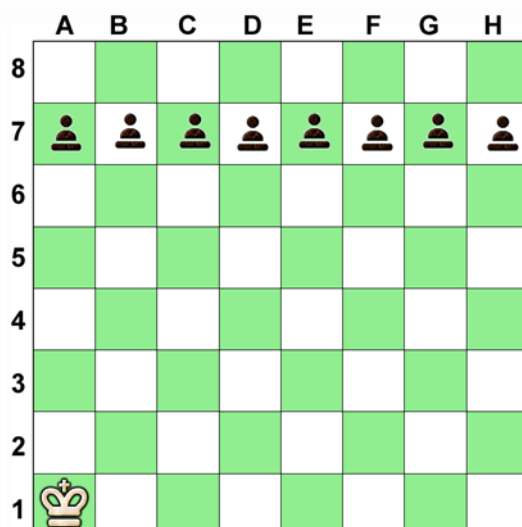
### Ejemplo 1. Hay Varias Rutas Seguras de Longitud Mínima

En este caso existen dos caminos de longitud mínima por los cuales el rey blanco puede transitar sin ser comido por ningún peón negro. Estos son

{A1, B2, C3, C4, D5, E6, F7, G8, H8}

{A1, B2, C3, D3, E4, F5, G6, H7, H8}

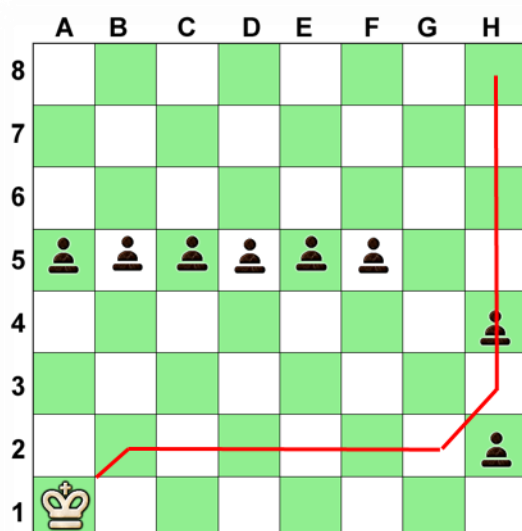
Su algoritmo debe retornar la longitud de cualquiera de estos caminos, es decir 9.



### Ejemplo 2. No Hay Ruta Segura

No existe una ruta segura para el rey blanco. Los peones negros bloquean el acceso a la casilla H8.

En este caso su algoritmo debe retornar cero.



### Ejemplo 3. Hay Ruta Segura (comiendo peones)

Existe una ruta segura, pero el rey debe comerse un peón para poder pasar sin ser comido. En este caso, la ruta de longitud mínima es

{A1, B2, C2, D2, E2, F2, G2, H3, H4, H5, H6, H7, H8}

Su algoritmo debe retornar la longitud de este camino, es decir 13.

#### Aclaraciones:

- Algunos casos, como el Ejemplo 2, implican la verificación de muchos caminos antes de poder decir que no hay ruta segura. **Puede ocurrir que su algoritmo tarde demasiado en dar respuesta.** Casos como este no serán la tónica de la evaluación, pero estarán presentes para comprobar la optimalidad de su algoritmo.
- Su algoritmo no deberá exceder un tiempo de ejecución de 10 seg. De ocurrir, la respuesta se considerará incorrecta.
- Puede ocurrir que la cantidad de peones en los tableros sea distinta de 8.