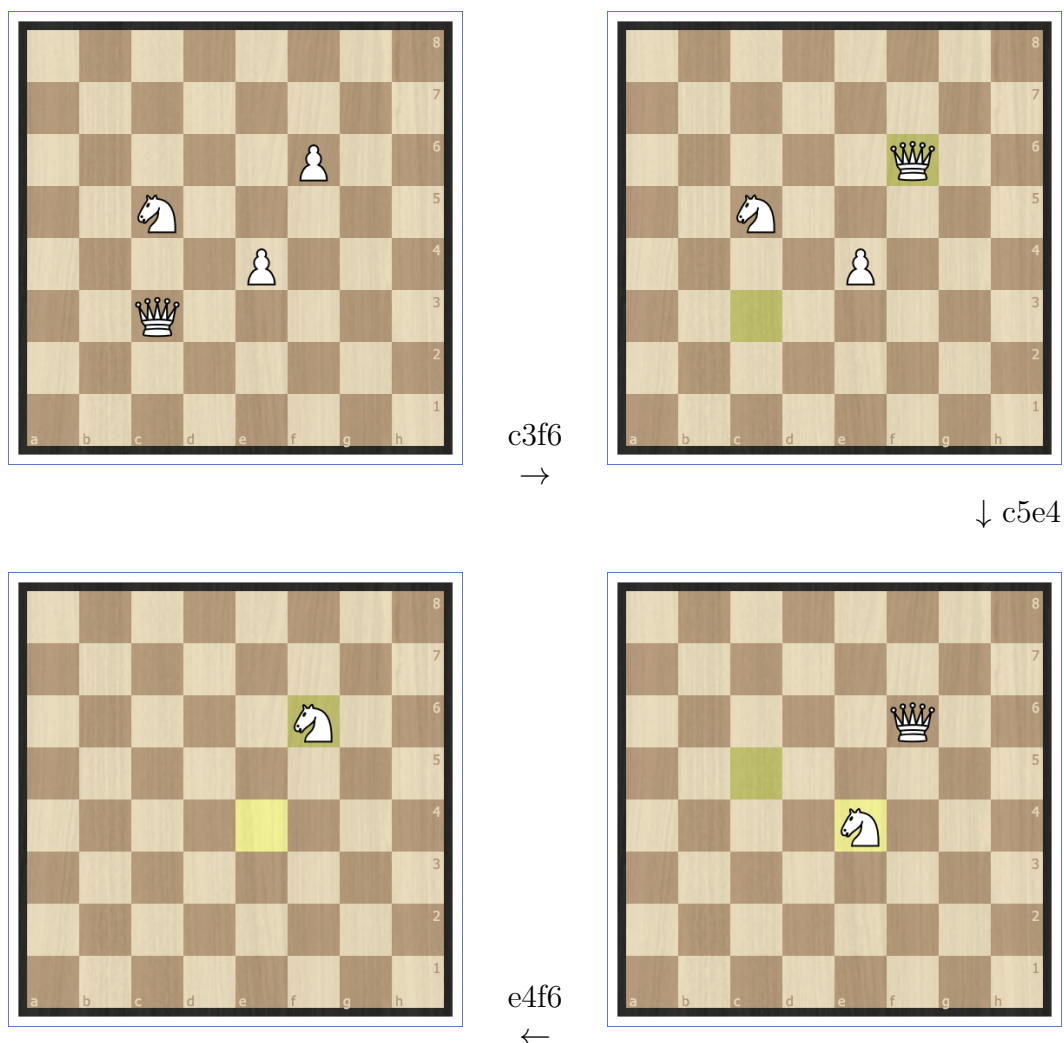


## Prueba Práctica Competitiva: Ajedrez Solitario

Se trata de escribir un solucionador para un puzzle al que llamaremos “Ajedrez Solitario”, que consiste en lo siguiente. En un tablero de ajedrez estándar (es decir, de dimensiones  $8 \times 8$ ) se disponen una serie de piezas blancas. Puede haber cualquier número de ellas, incluso más de un rey, más de una dama, más de dos torres, etc. El puzzle tendrá solución si se encuentra una secuencia de movimientos de captura que nos permita llegar a tener una sola pieza en el tablero. Dicha pieza no tiene porqué ser un rey. Podría ser cualquier otra pieza. En resumen, las reglas son simples:

- Las piezas se mueven como las piezas estándar de ajedrez. Únicamente tiene que tener en cuenta que se trata de piezas blancas, con lo cual, los peones sólo pueden avanzar, nunca retroceder. En términos de las figuras de ejemplo que aparecen más abajo, los peones sólo pueden ir hacia arriba, nunca hacia abajo.
- Sólo se pueden realizar movimientos de captura.
- Está permitido capturar al rey y continuar jugando.
- El objetivo es finalizar con una sola pieza en el tablero.

A continuación se muestra un ejemplo de partida:



En este caso, en el primer movimiento, la dama en c3 come al peón en f6. Seguidamente, el caballo en c5 come al peón en e4. Y por último, el caballo que ahora está en e4 come a la dama que ahora estaba en f6, y la partida finaliza con éxito.

Como se puede observar, no se especifica cuál es la pieza que debe empezar la secuencia de movimientos. Podría ser cualquiera. Y, por supuesto, la primera pieza que come no tiene porqué ser la única que lo haga. A lo largo de la secuencia de movimientos podrían existir varias piezas que efectúen movimientos de captura. De hecho, en el ejemplo anterior, hay dos: la dama y el caballo. Precisamente, en estos dos aspectos radica la dificultad del ejercicio. Finalmente, debe quedar claro también que pueden existir tableros sin solución, y tableros cuya solución no sea única.

Escriba un programa `ajedrez.ml`, que es el único que deberá entregar después. Supongamos que a partir de este programa obtenemos un ejecutable `ajedrez`. Su comportamiento debe ser el siguiente. Leerá un string desde la línea de comandos, donde vendrá especificada la configuración inicial del tablero: cada tres caracteres de este string corresponderán a una pieza; el primer carácter será `r` para el rey, `d` para la dama, `t` para la torre, `a` para el alfil, `c` para el caballo o `p` para el peón; el segundo carácter será la coordenada  $x$  de la pieza en formato letra de la `a` a la `h`; y el tercer carácter será la coordenada  $y$  de la pieza en formato número del 1 al 8. Si el puzzle correspondiente a este tablero no tiene solución, se escribirá una línea con la palabra `imposible`. Y si tiene solución, se escribirá una línea con la secuencia de movimientos. Cada movimiento constará de cuatro caracteres: las dos coordenadas de la pieza que captura, seguidas de las dos coordenadas de la pieza capturada. Cada movimiento irá separado del anterior mediante un espacio en blanco.

A continuación se muestra el ejemplo de ejecución correspondiente al tablero de la primera página:

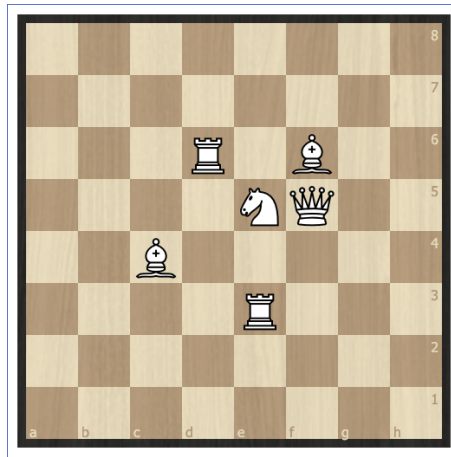
```
$ ./ajedrez dc3cc5pe4pf6  
c3f6 c5e4 e4f6
```

Al final de este documento se muestran otros ejemplos de ejecución correspondientes a tableros que también tienen solución. Tal y como se ha indicado previamente, tenga en cuenta que la solución podría no ser única.

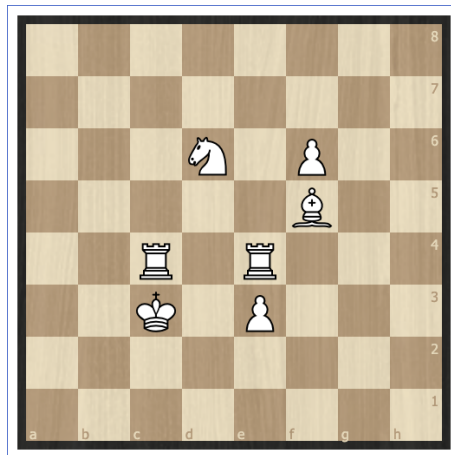
Esta práctica es competitiva, lo que significa que sólo unos pocos alumnos pueden puntuar con ella. Es importante mantener la discreción durante todo el proceso. Las implementaciones que se aprecien duplicadas serán descalificadas. Para ser admitido, el programa debe manejar tableros con hasta 30 piezas en tiempos del orden de pocos segundos.

Esta prueba práctica es de carácter individual. Sólo se admitirán las primeras 20 implementaciones correctas recibidas antes de la fecha límite de entrega. De estas, serán seleccionadas para su evaluación las 10 más eficientes. El código debe venir debidamente comentado para su fácil comprensión. Para la calificación se tendrá en cuenta, además de la eficiencia, la sencillez, claridad y originalidad del código. La puntuación máxima para esta prueba es de 1,2 puntos de los 2 que se pueden obtener en el apartado de “Pruebas Prácticas”. Si se detecta algún plagio, los implicados en el mismo perderán la oportunidad de recibir ninguna puntuación en este apartado.

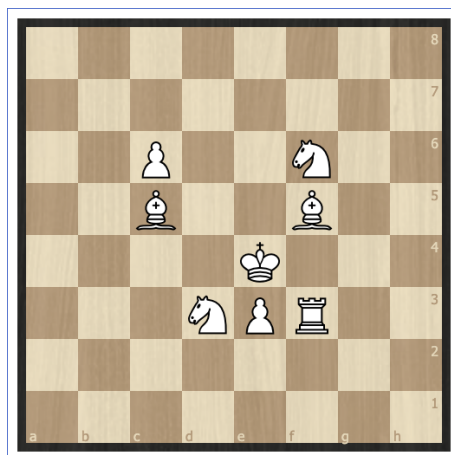
La entrega se realizará a través de la tarea en Moodle creada específicamente para este fin. Se entregará sólo un archivo de nombre `ajedrez.ml` con el código fuente del programa. La tarea permanecerá abierta hasta las 20:00 del 22 de diciembre, o hasta que se supere el número de 20 propuestas correctamente recibidas (lo que antes ocurra). Diariamente se actualizará el número de propuestas recibidas. La entrega requiere que se pulse el botón de “Enviar tarea”, y sólo se podrá realizar un envío. Es decir, una vez realizado el envío, la tarea ya no podrá editarse (no se podrá eliminar, ni se podrán realizar cambios en la misma).



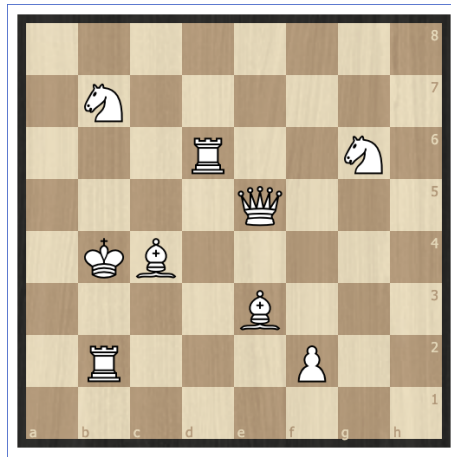
\$ ./ajedrez ac4td6te3ce5df5af6  
d6f6 f6f5 e5c4 c4e3 e3f5



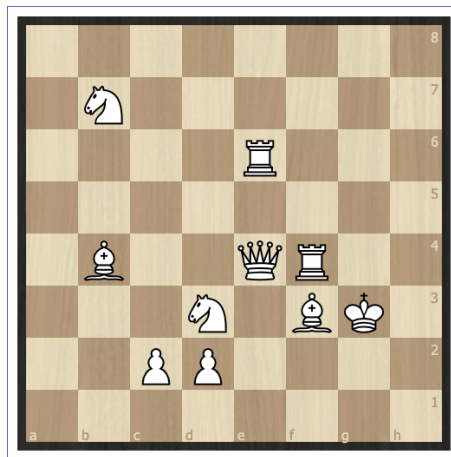
\$ ./ajedrez rc3tc4cd6pe3te4af5pf6  
c4c3 c3e3 e3e4 f5e4 d6e4 e4f6



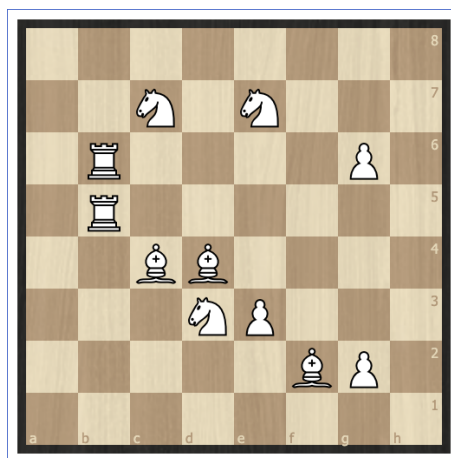
\$ ./ajedrez ac5pc6cd3pe3re4tf3af5cf6  
d3c5 c5e4 f3e3 e3e4 f6e4 f5e4 e4c6



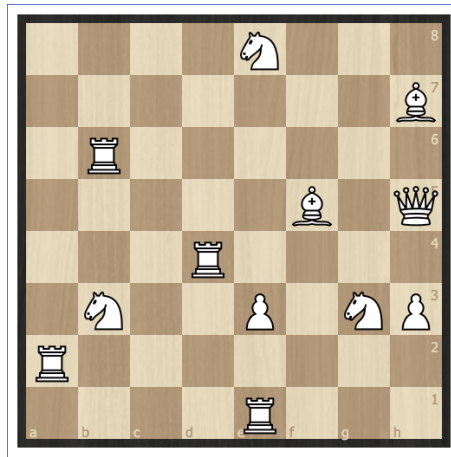
```
$ ./ajedrez tb2rb4cb7ac4td6ae3de5pf2cg6
b2b4 b4c4 b7d6 d6c4 g6e5 e5c4 c4e3 f2e3
```



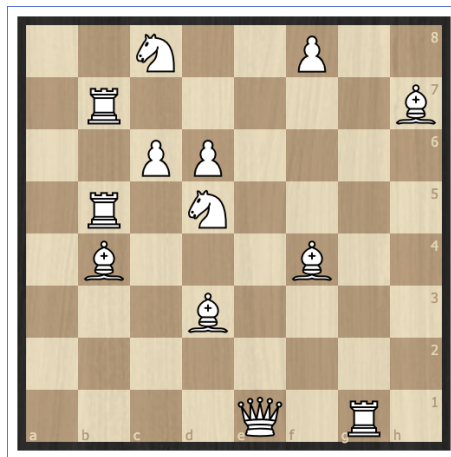
```
$ ./ajedrez ab4cb7pc2pd2cd3de4te6af3tf4rg3
b4d2 d2f4 c2d3 d3e4 e6e4 g3f4 f4e4 f3e4 e4b7
```



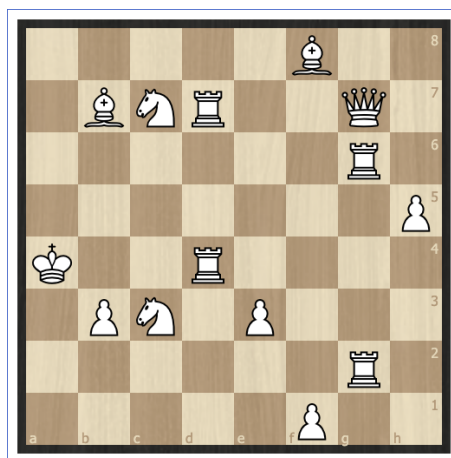
```
$ ./ajedrez tb5tb6ac4cc7cd3ad4pe3ce7af2pg2pg6
c7b5 c4b5 b5d3 d3g6 d4e3 e3f2 e7g6 b6g6 g6g2 g2f2
```



\$ ./ajedrez ta2cb3tb6td4te1pe3ce8af5cg3ph3dh5ah7  
b6b3 e1e3 h5e8 e8e3 h7f5 g3f5 f5d4 d4b3 e3h3 h3b3 b3a2



\$ ./ajedrez ab4tb5tb7pc6cc8ad3cd5pd6de1af4pf8tg1ah7  
c6b7 h7d3 g1e1 c8d6 f4d6 d6f8 f8b4 b5b7 b7b4 d5b4 b4d3 d3e1



\$ ./ajedrez ra4pb3ab7cc3cc7td4td7pe3pf1af8tg2tg6dg7ph5  
a4b3 d4d7 d7c7 c7c3 c3e3 e3b3 b3b7 f1g2 h5g6 f8g7 b7g7 g7g6 g6g2