



Tarea 3

Ajedrez

Entrega: miércoles 23 de mayo a las 21:00 hrs.

1. Tarea

El ajedrez es un juego de mesa para 2 personas, que está compuesto por un tablero cuadriculado de 64 espacios, y 16 piezas por lado. Se puede encontrar un resumen de las reglas en la siguiente pagina https://es.wikipedia.org/wiki/Ajedrez#Resumen_de_las_reglas_del_juego.

Usted deberá realizar una implementación de este juego, con las siguientes características:

- 1) El juego debe ser entre dos personas.
- 2) Al momento de ejecutarse debe indicar por el flag -t si se recibirán las instrucciones por consola o desde un archivo de texto, por ejemplo:

- Se reciben las instrucciones por consola:

```
./ajedrez
```

- Se reciben las instrucciones desde un archivo:

```
./ajedrez -t instrucciones.txt
```

Este archivo poseerá una instrucción por línea, y cada instrucción seguirá el mismo formato que si se estuviera leyendo por consola. Además, se aclara que la primera instrucción siempre corresponderá al jugador de las piezas blancas. Un ejemplo del archivo seria:

```
instrucciones.txt  
a2 a3  
c7 c6  
...
```



3) El tablero estará formado de la siguiente manera:

| | a | b | c | d | e | f | g | h | |
|---|----|----|----|----|----|----|----|----|---|
| 8 | a8 | b8 | c8 | d8 | e8 | f8 | g8 | h8 | 8 |
| 7 | a7 | b7 | c7 | d7 | e7 | f7 | g7 | h7 | 7 |
| 6 | a6 | b6 | c6 | d6 | e6 | f6 | g6 | h6 | 6 |
| 5 | a5 | b5 | c5 | d5 | e5 | f5 | g5 | h5 | 5 |
| 4 | a4 | b4 | c4 | d4 | e4 | f4 | g4 | h4 | 4 |
| 4 | a4 | b4 | c4 | d4 | e4 | f4 | g4 | h4 | 4 |
| 3 | a3 | b3 | c3 | d3 | e3 | f3 | g3 | h3 | 3 |
| 2 | a2 | b2 | c2 | d2 | e2 | f2 | g2 | h2 | 2 |
| 1 | a1 | b1 | c1 | d1 | e1 | f1 | g1 | h1 | 1 |
| | a | b | c | d | e | f | g | h | |

4) Las instrucciones para los movimientos serán:

`posición_inicial posición_final`

Como por ejemplo

a2 a3
b8 c6
etc...

Además, siempre se pedirá primero el movimiento del jugador blanco.

5) Si se entrega un movimiento invalido, se deberá pedir el movimiento de ese jugador hasta que se realice un movimiento válido



6) Las piezas estarán representadas por letras:

- **Peón:** P
- **Torre:** T
- **Alfil:** B
- **Caballo:** N
- **Reina:** Q
- **Rey:** K

Además estas deberán ir acompañadas de la letra **W** para indicar si es blanca, o la letra **D** para indicar si es negra.

7) Las piezas blancas siempre deberán partir en la parte inferior del tablero, y las negras en la parte superior.

8) El tablero al inicio de cada juego se debería ver de la siguiente manera:

| | a | b | c | d | e | f | g | h | |
|-------|----|----|----|----|----|----|----|----|---|
| ----- | | | | | | | | | |
| 8 | TD | ND | BD | KD | QD | BD | ND | TD | 8 |
| ----- | | | | | | | | | |
| 7 | PD | PD | PD | PD | PD | PD | PD | PD | 7 |
| ----- | | | | | | | | | |
| 6 | | | | | | | | | 6 |
| ----- | | | | | | | | | |
| 5 | | | | | | | | | 5 |
| ----- | | | | | | | | | |
| 4 | | | | | | | | | 4 |
| ----- | | | | | | | | | |
| 4 | | | | | | | | | 4 |
| ----- | | | | | | | | | |
| 3 | | | | | | | | | 3 |
| ----- | | | | | | | | | |
| 2 | PW | PW | PW | PW | PW | PW | PW | PW | 2 |
| ----- | | | | | | | | | |
| 1 | TW | NW | BW | KW | QW | BW | NW | TW | 1 |
| ----- | | | | | | | | | |
| | a | b | c | d | e | f | g | h | |



- 9) El juego terminará cuando no se puedan realizar mas movimientos válidos (empate), o algún jugador se coma el rey del contrincante. en cualquier caso se deberá imprimir el resultado
- 10) Su programa deberá componerse de los siguientes módulos (archivos):
- a) **main.cpp**: Archivo donde estará la función main de su programa
 - b) **io.cpp**: Archivo en donde se encontrarán todas las funciones relacionadas al input/output de su programa
 - c) **chess.cpp**: Archivo donde se manejará toda la lógica del juego de ajedrez.
 - d) Al menos un módulo más a su elección.

Cada módulo deberá estar acompañado de su respectivo archivo de encabezados **.h**. La separación del programa en estos archivos permite escalabilidad a futuro, permitiendo adaptar el programa a interfaces gráficas sin tener que reescribir la parte importante del programa (la lógica), entre otras cosas.

2. Bonus

Si desea que su tarea tenga nota máxima 8, deberá crear hacer que su programa controle a un jugador mediante una inteligencia artificial (IA) básica/mínima. Deberá explicar su funcionamiento en un archivo aparte llamado **ia.txt**. Además, la IA deberá estar en un módulo **ia.cpp**, y si el usuario desea jugar contra la IA deberá indicarlo mediante el flag **-i**.

3. Consideraciones Generales

Su programa debe ser robusto frente a datos que no corresponden. Esto quiere decir que su programa no debe caerse en caso de que el usuario haga operaciones matemáticas no válidas (p.ej., jugadas inválidas, coordenadas inexistentes, etc.). Sin embargo, pueden asumir que el usuario será amigable, en el sentido de que cuando le soliciten un número este no ingresará letras, por ejemplo.

4. Evaluación y Entrega

Formato de entrega: Subir un solo archivo comprimido con todo el código fuente de su programa al módulo de tareas de la página del curso en <https://saf.uandes.cl/ing/>, con el nombre de archivo “NOMBRE_APELLIDO_Tarea3.tar.gz” el cual deberá contener los archivos de código fuente en C (archivos **.c** y **.h**), reemplazando “NOMBRE” y “APELLIDO” según corresponda (ej. ALAN_TURING_Tarea1.tar.gz). Los archivos compilados no serán tomados en cuenta, si se llega a subir solo un archivo compilado, este será ignorado, y será evaluado con nota 1.

En el momento de compilación, se requiere incluir las siguientes flags al compilador:



`-Wall -Wextra -Wundef -Werror -Wuninitialized -Winit-self`

Junto a los archivos de código base, debe venir incluido un archivo **Makefile** el cual permita realizar la compilación de manera automática con el comando **make**, este debe incluir las siguientes flags:

`-Wall -Wextra -Wundef -Werror -Wuninitialized -Winit-self`

Al compilar se debe generar un ejecutable llamado “tarea_3”. Si este paso falla y se debe realizar la compilación de forma manual, o el ejecutable no se llama como se indica se procederá a descontar 1 punto de la nota obtenida.

Aquellas tareas que no compilen de la forma normal (`gcc -FLAGS -o salida entrada`) serán evaluadas con nota 1. Las que compilen, pero se caigan durante la ejecución, serán evaluadas con nota máxima 3.

Su programa será evaluado con múltiples casos de prueba y deberá ser capaz de ejecutarlos todos de manera correcta. De fallar en algún caso de prueba serán descontados los puntos correspondientes a dicho caso.

La entrega atrasada será evaluada con una penalización de un punto en la nota final por cada hora de atraso o fracción (aunque sea ésta de menos de un minuto).

5. Consideraciones de Trabajo

El trabajo en esta tarea deberá hacerse individual o en parejas. Cuide su tarea para que no sea copiada parcial o íntegramente por otros. Todas las tareas entregadas serán comparadas por un sistema automático de detección de plagios. Cualquier copia será penalizada, recibiendo el mismo castigo tanto quien copia como quien permite que le copien. También es considerada copia cualquier ayuda externa recibida directamente en la tarea, sin importar si proviene de un alumno del curso, de la universidad, o de otro lugar. El castigo será establecido por el Consejo de la Facultad, siendo como mínimo un 1,0 de promedio en el curso.