Trabajo práctico Final

1 Presentación del ejercicio

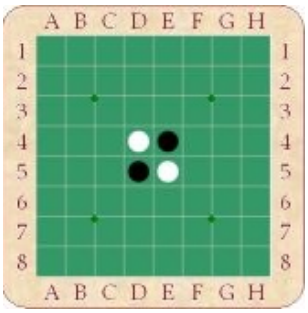
Queremos realizar un programa que permita jugar al juego Othello.

2 Características del juego

Othello es un juego de mesa que se desarrolla sobre un tablero de 8×8 con 64 fichas bicolores (un color para cada jugador). El juego finaliza cuando ya no pueden agregarse fichas y gana el jugador cuyo color tiene más fichas en el tablero. Puede haber empate en caso de que ninguno pueda agregar fichas y ambos tengan la misma cantidad.

3 Desarrollo del juego

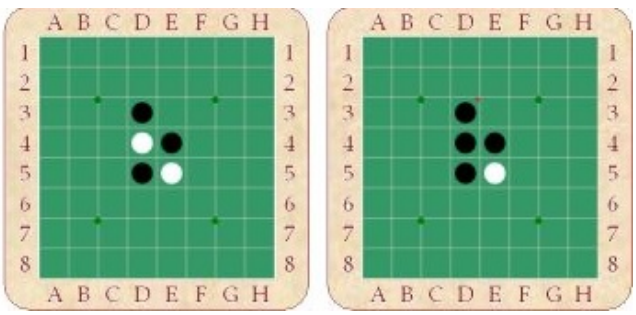
El juego comienza con esta configuración inicial:



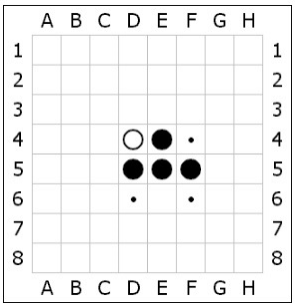
A partir de la misma se sortea el inicio. Supongamos que comienza el jugador con fichas negras.

La única restricción que existe al momento de poner una ficha es que sólo se pueden poner fichas en casillas que estén en contacto con una ficha del otro color y, además, que con ese movimiento se encierre una o más fichas del adversario entre la ficha que se acaba de poner en el tablero y otra ficha de jugador. Al realizarse el movimiento, las fichas del adversario que han quedado encerradas se vuelven del color de las fichas del jugador que acaba de utilizar su turno.
En caso de que no exista una posición que cumpla con esas características el jugador debe pasar el turno.

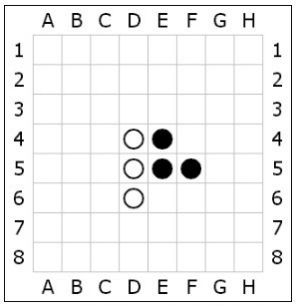
Ahora bien, para comenzar el juego, el jugador con fichas negras puede poner una ficha en $C4$, $D3$, $E6$ o $F5$.
Supongamos que ubica la ficha en $D3$. Entonces, como consecuencia de la jugada se tendría:



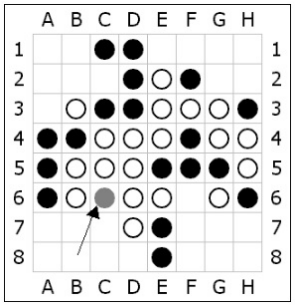
Veamos algunos ejemplos para terminar de entender la dinámica del juego.
En la siguiente configuración de fichas, el jugador con fichas blancas puede poner una ficha en $F4$, $F6$ o $D6$.



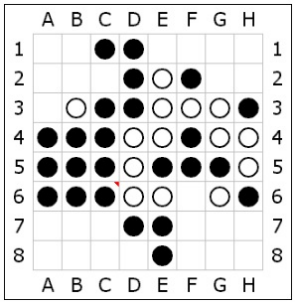
Si juega en $D6$ el tablero queda:



Ahora, analicemos un tablero más complejo.



Si el jugador con fichas negras pone una ficha en $C6$ genera cambios en forma vertical, horizonal y diagonal. No se voltean ni $D6$ ni $E6$ debido a la casilla vacía en $F6$. Es una buena jugada porque además amenaza la diagonal ya que poniendo una ficha en $G2$ daría vuelta a las fichas $F3$, $E4$ y $E5$.



4 Programa en C

El programa en C tomará como dato de entrada un archivo con la siguiente información:

- nombres de los jugadores;
- color asignado a cada jugador;
- color que comenzó;
- jugadas realizadas.

4.1 Ejemplo de entrada

Supongamos que tenemos el siguiente archivo de entrada:

```
Alejandro,B
Federico,N
B
D6
C4
G5
```

Se puede ver que las dos primeras líneas indican el nombre de cada jugador y el color que tienen: B ó N. La siguiente línea indica qué color comienza. Luego de esto, en cada línea hay un movimiento que corresponde al color.

Observación 1: En caso de que un jugador tenga que pasar la línea estará vacía.
Observación 2: El formato del archivo **NO** puede ser modificado.

4.2 Funcionamiento del programa a entregar

El programa deberá:

a) Tomar un archivo de entrada y procesarlo;

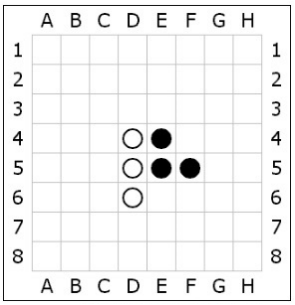
b) en caso que alguna de las jugadas encontradas en el archivo sea incorrecta debe indicarlo marcando qué jugador realizó esa jugada, mostrar el tablero hasta esa jugada (sin incluirla) y finalizar.

c) si al finalizar el procesamiento del archivo el tablero está completo (el juego terminó) se debe indicar quién ganó la partida;

d) en caso de no estar finalizada la jugada, se debe volcar el tablero resultante hasta ese momento en un archivo. El formato del mismo será el siguiente:

- si la casilla está vacía se escribe una X;
- si en la casilla hay una ficha blanca se escribe B;
- si en la casilla hay una ficha negra se escribe N;
- se escribe un caracter al lado del otro y cada fila del tablero es una línea (no hay líneas vacías entre ellas);
- en la línea siguiente se debe indicar qué color continúa jugando.

Vamos a ilustrar con un ejemplo. Si luego de procesar el archivo de entrada, el tablero resultante es:



el archivo que se debe escribir sería:

```
XXXXXXXX
XXXXXXXX
XXXXXXXX
XXXBNXXX
XXXBNNXX
XXXBXXXX
XXXXXXXX
XXXXXXXX
N
```

Observación: los nombres de los archivos se deben pasar como argumentos a la función `main`.

5 Programa en Python

El programa tomará como entrada el archivo generado por el programa en *C* y el color con el que jugará el jugador (el otro color será el que le corresponda a nuestro programa). Para comenzar a jugar, se deberá imprimir el tablero.

La dinámica será la siguiente:

- cuando le corresponda jugar al jugador, se deberá pedir que ingrese la jugada y validarla. Si la misma es incorrecta deberá pedir que reingrese la jugada hasta que la jugada ingresada sea válida. En caso contrario, se imprime el tablero resultante (luego de la jugada).
- cuando le corresponda jugar al programa deberá imprimir la jugada que eligió y el tablero resultante luego de la misma.

Nuestro programa tendrá dos niveles de juego:

- **Nivel 0**: el programa debe determinar cuáles son las jugadas posibles y *en forma aleatoria* elegir una de ellas;
- **Nivel 1**: el programa elige la jugada a realizar analizando cuál de ellas genera mayores cambios en el tablero (da vuelta más fichas del contrario).

Observación: Cuando se llama al programa se le pasa el nombre del archivo con la partida a cargar, el color con el que quiere jugar y el nivel del programa como argumentos.

6 Un poco de ayuda

A lo largo del trabajo se pide en distintos momentos, tanto en *Python* como en *C*, que se verifique la validez de una jugada. Dado que una jugada es válida sólo si al realizarse se generan ciertos cambios en el tablero, la forma más sencilla de saber si es válido ubicar una ficha en cierta posición es simular que se coloca esa ficha, realizar el proceso que se debería hacer para ejecutar la jugada y evaluar si luego de esto hubo cambios en el tablero o no. Si hubo cambios es porque esa posición es válida para jugar. Si no, no lo es.

7 Características de la entrega

Se pide que escriba un programa que cumpla con los siguientes requisitos:

- los nombres de funciones y variables deber ser significativos (estar relacionados con su uso);
- no se pueden usar variables globales, definidas fuera de funciones;
- se debe comentar, adecuadamente, el código;
- deben estar testeadas, adecuadamente, todas las funciones que se puedan. Tanto en Python como en C.

Además del código y los tests, se pide que entregue un informe explicando brevemente cómo fue el proceso de resolución del trabajo, cuáles fueron las decisiones más importantes que se tomaron y por qué.