Objective:

- Object's initialization.
- And a bit of logic as always to keep your brains working ☺

Challenge: *Reversi* (39)

https://en.wikipedia.org/wiki/Reversi
https://m.twoplayergames.org/play/reversi.html

Gear up guys you have to make a game named as Reversi ☺

Basic rules of Game:

This game has two players each will control each color (i.e black or white). Each reversi piece has a black side and a white side. On your turn, you place one piece on the board with your color facing up. You must place the piece so that an opponent's piece, or a row of opponent's pieces, is flanked by your pieces. All of the opponent's pieces between your pieces are then turned over to become your color.

Aim of the game

The object of the game is to own more pieces than your opponent when the game is over. The game is over when neither player has a move. Usually, this means the board is full.

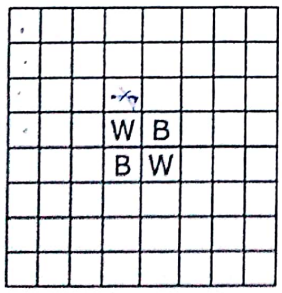
Moves Guide:

A move consists in placing from outside one piece on the board. Placed pieces can never be moved to another square later in the game.
The incorporation of the pieces must be made according to the following rules:

- The new piece must outflank one or more of the opponents placed pieces
- To outflank means that a single piece or one straight row (vertical, horizontal or diagonal) of pieces of the opponent is in both sides next to own pieces, with no empty squares between all those pieces
- The player who makes the move turns the outflanked pieces over, becoming all of them in own pieces
- If there is more than one outflanked row, all the involved pieces in those rows have to be flipped
- If it´s not possible to make this kind of move, turn is forfeited and the opponent repeats another move

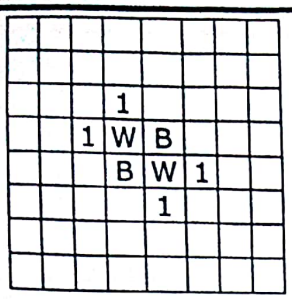
Start of the game

At start of the game Reversi board looks like this

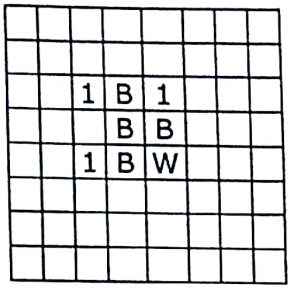


constructor should initialize board like this. You can use character 'B' in place of black color and 'W' in place of White color.

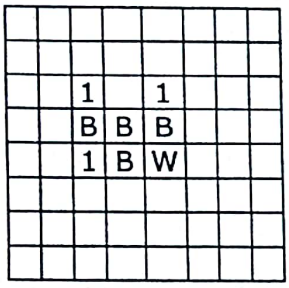
Just suppose the first turn is of Player-1 which has black color reversi piece.
So, what will the possible places? on which he can place his next piece for flanking the opposite player's pieces.



Yes, these four blocks with cross in it are the blocks which are valid moves for the black one to take. These are the POSSIBLE MOVES with which black piece can flank white pieces.



Just assume if u place your piece on (3,4) position. After this move the board will look like this.



Now assume you placed the piece on position (4,3) instead of (3,4) then the board will look like this.

And so on..... ☺

End of the game

The game ends when:

- One player wins, by making his color dominant on the board (one's count is greater than other and board is full).
- When the board is full and count of both pieces are equal (Tie/Draw).

Classes for the Game

ReversiBoard.h	
`enum GameStatus {WHITE_WIN='W', BLACK_WIN='B', IN_PROGRESS='P', DRAW='D'};`	
`enum PlayerTurn {BLACK_PLAYER='B', WHITE_PLAYER='W'};`	
`enum PlayerSymbol {BLACK_PLAYER_SYMBOL='B', WHITE_PLAYER_SYMBOL='W'};`	
`#define ROWS 8`	
`#define COLS 8`	
`class ReversiBoard`	
`{`	
`char data[ROWS][COLS];`	Represents Game Board
`PlayerTurn currentPlayer = BLACK_PLAYER;`	
`GameStatus gStatus = IN_PROGRESS;`	
`int whiteScore = 2;`	
`int blackScore = 2;`	
`public:`	
`ReversiBoard();`	Initializes the Board
`PlayerTurn getCurrentPlayer();`	Returns the current player
`void switchPlayerTurn();`	Switch the current Player Turn by updating currentPlayer.
`int placeDisc(int r, int c);`	Place Disc at the given row, col

	and also invert all in the way. Return 0 if given r,c are not on game board Return -1 if the r,c are on board but not valid move Return 1 if successful.
void displayBoard();	Print the board on screen
int getWhiteScore();	Return the score of White Player
int getBlackScore();	Return the score of Black Player
GameStatus getGameStatus();	Return the game status stored in gStatus
bool isMovePossible();	Return true if any valid move remains for the current player otherwise false.
};	

ReversiGame.h	
class ReversiGame	
{	
public:	
void playGame();	It's the game controller. Deals with the interfacing and handling of the Game.
};	

Game Interface/Sample Run

```
    1   2   3   4   5   6   7   8
  ---------------------------------
1|  .   .   .   .   .   .   .   .
2|  .   .   .   .   .   .   .   .
3|  .   .   .   1   .   .   .   .
4|  .   .   1   W   B   .   .   .
5|  .   .   .   B   W   1   .   .
6|  .   .   .   .   1   .   .   .
7|  .   .   .   .   .   .   .   .
8|  .   .   .   .   .   .   .   .

Black Score : 2
White Score : 2
player B enter position (row column): 6 5

    1   2   3   4   5   6   7   8
  ---------------------------------
1|  .   .   .   .   .   .   .   .
2|  .   .   .   .   .   .   .   .
3|  .   .   .   .   .   .   .   .
4|  .   .   .   W   B   1   .   .
5|  .   .   .   B   B   .   .   .
6|  .   .   .   1   B   1   .   .
7|  .   .   .   .   .   .   .   .
8|  .   .   .   .   .   .   .   .

Black Score : 4
White Score : 1
player W enter position (row column): 6 6

    1   2   3   4   5   6   7   8
  ---------------------------------
1|  .   .   .   .   .   .   .   .
2|  .   .   .   .   .   .   .   .
3|  .   .   1   1   .   .   .   .
4|  .   .   1   W   B   .   .   .
5|  .   .   1   B   W   1   .   .
6|  .   .   .   .   B   W   1   .
7|  .   .   .   .   .   .   .   .
8|  .   .   .   .   .   .   .   .
```

```
Black Score : 3
White Score : 3
player B enter position (row column): 5 6

     1   2   3   4   5   6   7   8
   ---------------------------------
1|   .   .   .   .   .   .   .   .
2|   .   .   .   .   .   .   .   .
3|   .   .   .   .   .   .   .   .
4|   .   .   .   W   B   2   .   .
5|   .   .   .   B   B   B   .   .
6|   .   .   .   2   B   W   .   .
7|   .   .   .   .   .   .   .   .
8|   .   .   .   .   .   .   .   .

Black Score : 5
White Score : 2
player W enter position (row column): 4 6

     1   2   3   4   5   6   7   8
   ---------------------------------
1|   .   .   .   .   .   .   .   .
2|   .   .   .   .   .   .   .   .
3|   .   .   1   1   1   1   1   .
4|   .   .   .   W   W   W   1   .
5|   .   .   .   B   B   W   1   .
6|   .   .   .   .   B   W   1   .
7|   .   .   .   .   .   .   1   .
8|   .   .   .   .   .   .   .   .

Black Score : 3
White Score : 5
player B enter position (row column):
```

Losers quit when they fail.

Winners fail until they succeed.

-- Robert kiyosaki --