Problem 3274: Check if Two Chessboard Squares Have the Same Color

Problem Information

Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given two strings,

coordinate1

and

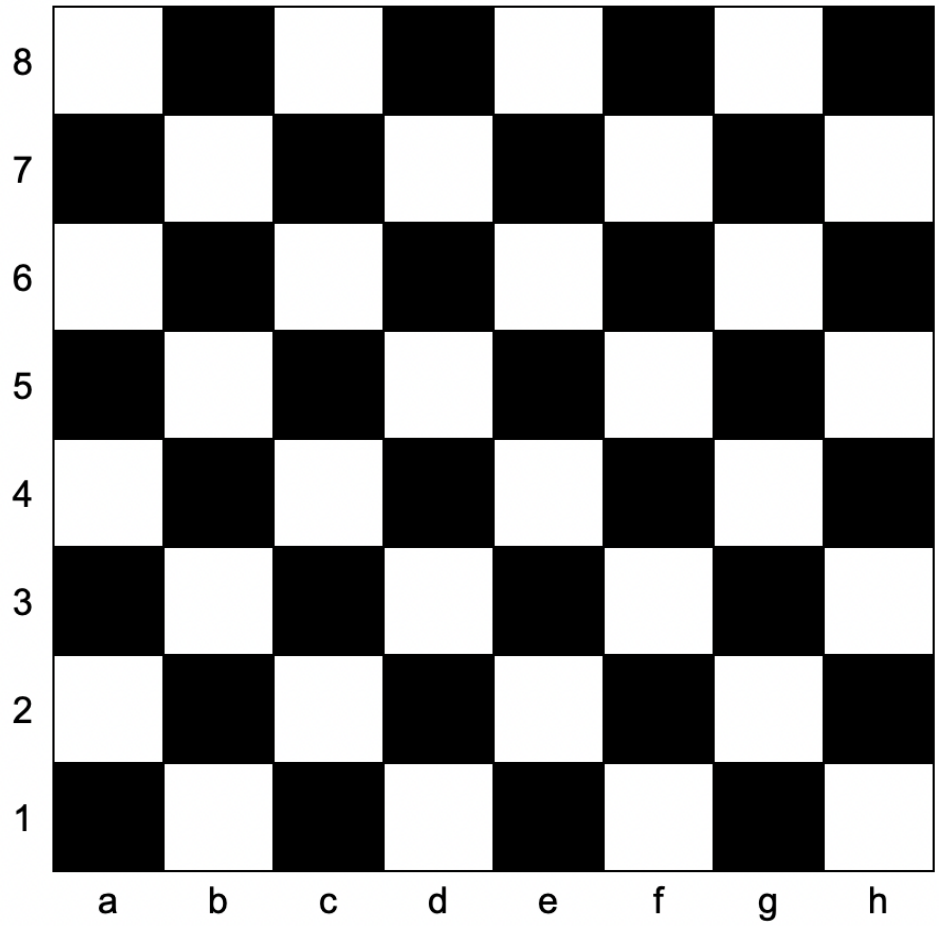
coordinate2

, representing the coordinates of a square on an

8 x 8

chessboard.

Below is the chessboard for reference.



Return

true

if these two squares have the same color and

false

otherwise.

The coordinate will always represent a valid chessboard square. The coordinate will always have the letter first (indicating its column), and the number second (indicating its row).

Example 1:

Input:

coordinate1 = "a1", coordinate2 = "c3"

Output:

true

Explanation:

Both squares are black.

Example 2:

Input:

coordinate1 = "a1", coordinate2 = "h3"

Output:

false

Explanation:

Square

"a1"

is black and

"h3"

is white.

Constraints:

coordinate1.length == coordinate2.length == 2

'a' <= coordinate1[0], coordinate2[0] <= 'h'

'1' <= coordinate1[1], coordinate2[1] <= '8'

Code Snippets

C++:

```
class Solution {
public:
bool checkTwoChessboards(string coordinate1, string coordinate2) {

}
};
```

Java:

```
class Solution {
public boolean checkTwoChessboards(String coordinate1, String coordinate2) {

}
}
```

Python3:

```
class Solution:
def checkTwoChessboards(self, coordinate1: str, coordinate2: str) -> bool:
```

Python:

```
class Solution(object):
def checkTwoChessboards(self, coordinate1, coordinate2):
"""
:type coordinate1: str
:type coordinate2: str
:rtype: bool
"""
```

JavaScript:

```
/**
* @param {string} coordinate1
* @param {string} coordinate2
* @return {boolean}
*/
var checkTwoChessboards = function(coordinate1, coordinate2) {
```

```
};
```

TypeScript:

```
function checkTwoChessboards(coordinate1: string, coordinate2: string):
boolean {

};
```

C#:

```
public class Solution {
public bool CheckTwoChessboards(string coordinate1, string coordinate2) {

}
}
```

C:

```
bool checkTwoChessboards(char* coordinate1, char* coordinate2) {

}
```

Go:

```
func checkTwoChessboards(coordinate1 string, coordinate2 string) bool {

}
```

Kotlin:

```
class Solution {
fun checkTwoChessboards(coordinate1: String, coordinate2: String): Boolean {

}
}
```

Swift:

```
class Solution {
func checkTwoChessboards(_ coordinate1: String, _ coordinate2: String) ->
```

```
Bool {

}
}
```

Rust:

```
impl Solution {
pub fn check_two_chessboards(coordinate1: String, coordinate2: String) ->
bool {

}
}
```

Ruby:

```
# @param {String} coordinate1
# @param {String} coordinate2
# @return {Boolean}
def check_two_chessboards(coordinate1, coordinate2)

end
```

PHP:

```
class Solution {

/**
* @param String $coordinate1
* @param String $coordinate2
* @return Boolean
*/
function checkTwoChessboards($coordinate1, $coordinate2) {

}
}
```

Dart:

```
class Solution {
bool checkTwoChessboards(String coordinate1, String coordinate2) {

}
```

```
}
```

Scala:

```
object Solution {
def checkTwoChessboards(coordinate1: String, coordinate2: String): Boolean =
{

}
}
```

Elixir:

```
defmodule Solution do
@spec check_two_chessboards(coordinate1 :: String.t, coordinate2 :: String.t)
:: boolean
def check_two_chessboards(coordinate1, coordinate2) do

end
end
```

Erlang:

```
-spec check_two_chessboards(Coordinate1 :: unicode:unicode_binary(),
Coordinate2 :: unicode:unicode_binary()) -> boolean().
check_two_chessboards(Coordinate1, Coordinate2) ->
.
```

Racket:

```
(define/contract (check-two-chessboards coordinate1 coordinate2)
(-> string? string? boolean?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Check if Two Chessboard Squares Have the Same Color
```

```
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public:
bool checkTwoChessboards(string coordinate1, string coordinate2) {

}
};
```

Java Solution:

```
/**
* Problem: Check if Two Chessboard Squares Have the Same Color
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

class Solution {
public boolean checkTwoChessboards(String coordinate1, String coordinate2) {

}
}
```

Python3 Solution:

```
"""
Problem: Check if Two Chessboard Squares Have the Same Color
Difficulty: Easy
Tags: string, math

Approach: String manipulation with hash map or two pointers
```

```
Time Complexity: O(n) or O(n log n)
Space Complexity: O(1) to O(n) depending on approach
"""

class Solution:
def checkTwoChessboards(self, coordinate1: str, coordinate2: str) -> bool:
# TODO: Implement optimized solution
pass
```

Python Solution:

```
class Solution(object):
def checkTwoChessboards(self, coordinate1, coordinate2):
"""
:type coordinate1: str
:type coordinate2: str
:rtype: bool
"""
```

JavaScript Solution:

```
/**
* Problem: Check if Two Chessboard Squares Have the Same Color
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

/**
* @param {string} coordinate1
* @param {string} coordinate2
* @return {boolean}
*/
var checkTwoChessboards = function(coordinate1, coordinate2) {

};
```

TypeScript Solution:

```
/**
* Problem: Check if Two Chessboard Squares Have the Same Color
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

function checkTwoChessboards(coordinate1: string, coordinate2: string):
boolean {

};
```

C# Solution:

```
/*
* Problem: Check if Two Chessboard Squares Have the Same Color
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
* Space Complexity: O(1) to O(n) depending on approach
*/

public class Solution {
public bool CheckTwoChessboards(string coordinate1, string coordinate2) {

}
}
```

C Solution:

```
/*
* Problem: Check if Two Chessboard Squares Have the Same Color
* Difficulty: Easy
* Tags: string, math
*
* Approach: String manipulation with hash map or two pointers
* Time Complexity: O(n) or O(n log n)
```

```
* Space Complexity: O(1) to O(n) depending on approach
*/

bool checkTwoChessboards(char* coordinate1, char* coordinate2) {

}
```

Go Solution:

```
// Problem: Check if Two Chessboard Squares Have the Same Color
// Difficulty: Easy
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

func checkTwoChessboards(coordinate1 string, coordinate2 string) bool {

}
```

Kotlin Solution:

```
class Solution {
fun checkTwoChessboards(coordinate1: String, coordinate2: String): Boolean {

}
}
```

Swift Solution:

```
class Solution {
func checkTwoChessboards(_ coordinate1: String, _ coordinate2: String) ->
Bool {

}
}
```

Rust Solution:

```
// Problem: Check if Two Chessboard Squares Have the Same Color
// Difficulty: Easy
// Tags: string, math
//
// Approach: String manipulation with hash map or two pointers
// Time Complexity: O(n) or O(n log n)
// Space Complexity: O(1) to O(n) depending on approach

impl Solution {
pub fn check_two_chessboards(coordinate1: String, coordinate2: String) ->
bool {

}
}
```

Ruby Solution:

```
# @param {String} coordinate1
# @param {String} coordinate2
# @return {Boolean}
def check_two_chessboards(coordinate1, coordinate2)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $coordinate1
* @param String $coordinate2
* @return Boolean
*/
function checkTwoChessboards($coordinate1, $coordinate2) {

}
}
```

Dart Solution:

```
class Solution {
bool checkTwoChessboards(String coordinate1, String coordinate2) {
```

```
}
}
```

Scala Solution:

```
object Solution {
def checkTwoChessboards(coordinate1: String, coordinate2: String): Boolean =
{

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec check_two_chessboards(coordinate1 :: String.t, coordinate2 :: String.t)
:: boolean
def check_two_chessboards(coordinate1, coordinate2) do

end
end
```

Erlang Solution:

```
-spec check_two_chessboards(Coordinate1 :: unicode:unicode_binary(),
Coordinate2 :: unicode:unicode_binary()) -> boolean().
check_two_chessboards(Coordinate1, Coordinate2) ->
.
```

Racket Solution:

```
(define/contract (check-two-chessboards coordinate1 coordinate2)
(-> string? string? boolean?)
)
```