Problem 1812: Determine Color of a Chessboard Square

Problem Information

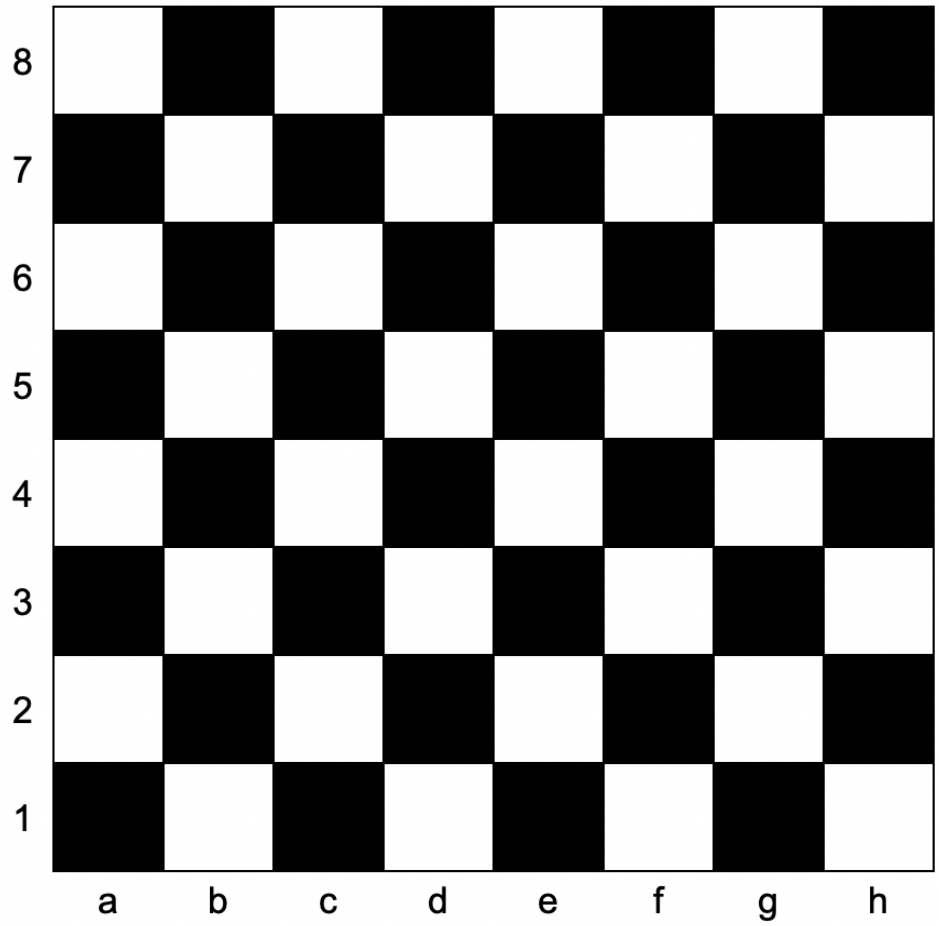
Difficulty: Easy
Acceptance Rate: 0.00%
Paid Only: No

Problem Description

You are given

coordinates

, a string that represents the coordinates of a square of the chessboard. Below is a chessboard for your reference.



Return

true

if the square is white, and

false

if the square is black

.

The coordinate will always represent a valid chessboard square. The coordinate will always have the letter first, and the number second.

Example 1:

Input:

coordinates = "a1"

Output:

false

Explanation:

From the chessboard above, the square with coordinates "a1" is black, so return false.

Example 2:

Input:

coordinates = "h3"

Output:

true

Explanation:

From the chessboard above, the square with coordinates "h3" is white, so return true.

Example 3:

Input:

coordinates = "c7"

Output:

false

Constraints:

coordinates.length == 2

'a' <= coordinates[0] <= 'h'

'1' <= coordinates[1] <= '8'

Code Snippets

C++:

```
class Solution {
public:
bool squareIsWhite(string coordinates) {

}
};
```

Java:

```
class Solution {
public boolean squareIsWhite(String coordinates) {

}
}
```

Python3:

```
class Solution:
def squareIsWhite(self, coordinates: str) -> bool:
```

Python:

```
class Solution(object):
def squareIsWhite(self, coordinates):
"""
:type coordinates: str
:rtype: bool
"""
```

JavaScript:

```
/**
* @param {string} coordinates
```

```
* @return {boolean}
*/
var squareIsWhite = function(coordinates) {

};
```

TypeScript:

```
function squareIsWhite(coordinates: string): boolean {

};
```

C#:

```
public class Solution {
public bool SquareIsWhite(string coordinates) {

}
}
```

C:

```
bool squareIsWhite(char* coordinates) {

}
```

Go:

```
func squareIsWhite(coordinates string) bool {

}
```

Kotlin:

```
class Solution {
fun squareIsWhite(coordinates: String): Boolean {

}
}
```

Swift:

```
class Solution {
func squareIsWhite(_ coordinates: String) -> Bool {

}
}
```

Rust:

```
impl Solution {
pub fn square_is_white(coordinates: String) -> bool {

}
}
```

Ruby:

```
# @param {String} coordinates
# @return {Boolean}
def square_is_white(coordinates)

end
```

PHP:

```
class Solution {

/**
* @param String $coordinates
* @return Boolean
*/
function squareIsWhite($coordinates) {

}
}
```

Dart:

```
class Solution {
bool squareIsWhite(String coordinates) {

}
}
```

Scala:

```
object Solution {
def squareIsWhite(coordinates: String): Boolean = {

}
}
```

Elixir:

```
defmodule Solution do
@spec square_is_white(coordinates :: String.t) :: boolean
def square_is_white(coordinates) do

end
end
```

Erlang:

```
-spec square_is_white(Coordinates :: unicode:unicode_binary()) -> boolean().
square_is_white(Coordinates) ->
.
```

Racket:

```
(define/contract (square-is-white coordinates)
(-> string? boolean?)
)
```

Solutions

C++ Solution:

```
/*
* Problem: Determine Color of a Chessboard Square
* 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 squareIsWhite(string coordinates) {

}
};
```

Java Solution:

```
/**
* Problem: Determine Color of a Chessboard Square
* 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 squareIsWhite(String coordinates) {

}
}
```

Python3 Solution:

```
"""
Problem: Determine Color of a Chessboard Square
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 squareIsWhite(self, coordinates: str) -> bool:
# TODO: Implement optimized solution
```

```
pass
```

Python Solution:

```
class Solution(object):
def squareIsWhite(self, coordinates):
"""
:type coordinates: str
:rtype: bool
"""
```

JavaScript Solution:

```
/**
* Problem: Determine Color of a Chessboard Square
* 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} coordinates
* @return {boolean}
*/
var squareIsWhite = function(coordinates) {

};
```

TypeScript Solution:

```
/**
* Problem: Determine Color of a Chessboard Square
* 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 squareIsWhite(coordinates: string): boolean {

};
```

C# Solution:

```
/*
* Problem: Determine Color of a Chessboard Square
* 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 SquareIsWhite(string coordinates) {

}
}
```

C Solution:

```
/*
* Problem: Determine Color of a Chessboard Square
* 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 squareIsWhite(char* coordinates) {

}
```

Go Solution:

```
// Problem: Determine Color of a Chessboard Square
// 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 squareIsWhite(coordinates string) bool {

}
```

Kotlin Solution:

```
class Solution {
fun squareIsWhite(coordinates: String): Boolean {

}
}
```

Swift Solution:

```
class Solution {
func squareIsWhite(_ coordinates: String) -> Bool {

}
}
```

Rust Solution:

```
// Problem: Determine Color of a Chessboard Square
// 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 square_is_white(coordinates: String) -> bool {

}
```

```
}
```

Ruby Solution:

```
# @param {String} coordinates
# @return {Boolean}
def square_is_white(coordinates)

end
```

PHP Solution:

```
class Solution {

/**
* @param String $coordinates
* @return Boolean
*/
function squareIsWhite($coordinates) {

}
}
```

Dart Solution:

```
class Solution {
bool squareIsWhite(String coordinates) {

}
}
```

Scala Solution:

```
object Solution {
def squareIsWhite(coordinates: String): Boolean = {

}
}
```

Elixir Solution:

```
defmodule Solution do
@spec square_is_white(coordinates :: String.t) :: boolean
def square_is_white(coordinates) do

end
end
```

Erlang Solution:

```
-spec square_is_white(Coordinates :: unicode:unicode_binary()) -> boolean().
square_is_white(Coordinates) ->
.
```

Racket Solution:

```
(define/contract (square-is-white coordinates)
(-> string? boolean?)
)
```