Solution Review 1: Check Divisibility by 3 and 4

This lesson gives a detailed solution review to the challenge in the previous lesson.

We'll cover the following

Solution:

```
fn test_divisibility_by_3_4(a:i32)->i32{
    //check if number is divisible by 3 and 4
    if a % 3 == 0 && a % 4 == 0{
        0
    }
    //check if number is divisible by 3 and not by 4
    else if a % 3 == 0 && a % 4 != 0 {
        1
    }
    //check if number is divisible not by 3 but 4
    else if a % 3 != 0 && a % 4 == 0 {
        2
    }
    //check if neither divisible by 3 nor 4
    else {
        -1
    }
}
fn main(){
    println!(" Number = 12 : {}", test_divisibility_by_3_4(12));
    println!(" Number = 9  : {}", test_divisibility_by_3_4(9));
    println!(" Number = 8  : {}", test_divisibility_by_3_4(8));
    println!(" Number = 23 : {}", test_divisibility_by_3_4(23));
}
```

Explanation

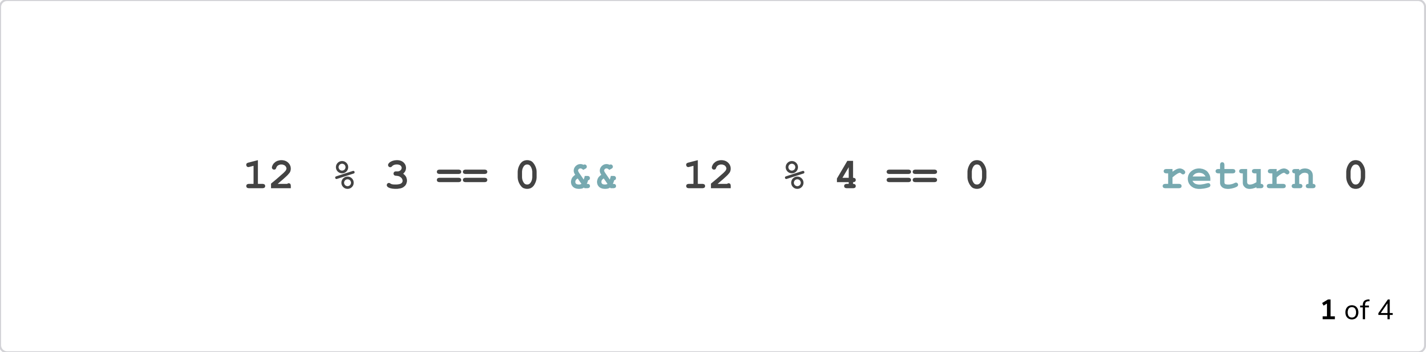
The `test_divisibility_by_3_4` takes an integer `a` **as a parameter** to the function and **returns** an integer of type `i32`.

- On **line 3**, the `if` condition checks if the number `a` is **divisible by 3 and 4**, it **returns** `0` on **line 4**.

returns `0` on **line 4**.

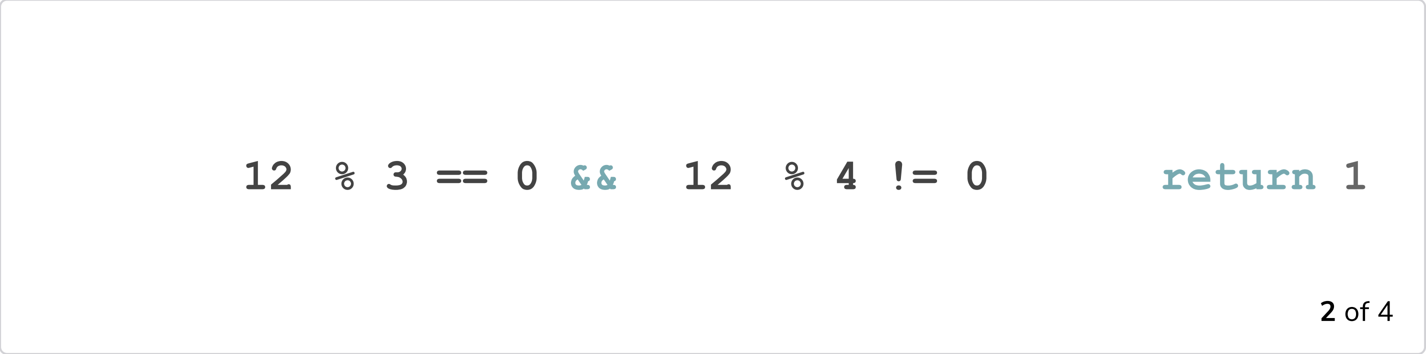
- On **line** 7, the `else if` executes if the `if` condition fails.
    - The condition checks if the number `a` is **divisible by 3 and not by 4**, it **returns** `1` on **line 8**.
- On **line 11**, the `else if` executes if the first `else if` condition fails.
    - The condition checks if the number `a` is **not divisible by 3 but by 4**, it **returns** `2` on **line 12**.
- On **line 15**, the `else`, executes if all the above conditions fail and **returns** `-1` on **line 16**.

The following illustration shows all the four conditions that the above code tests:



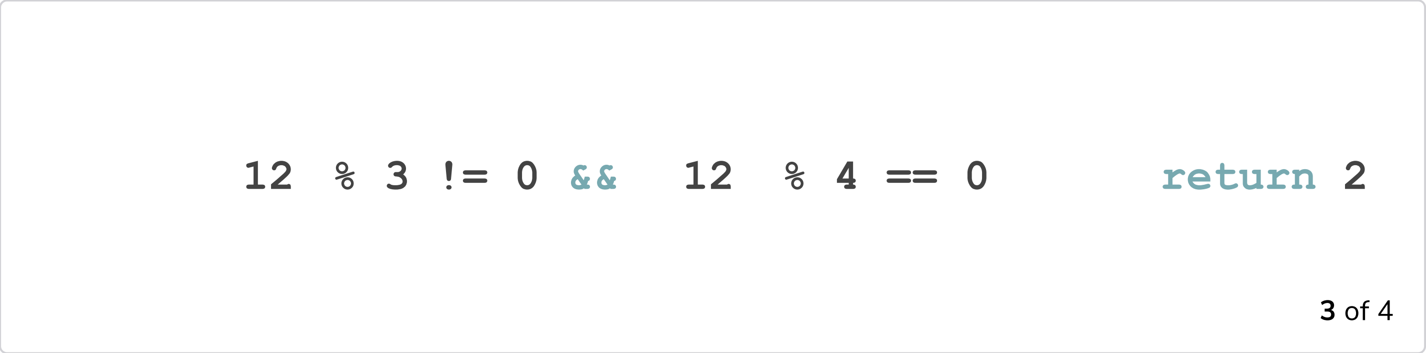
```
12 % 3 == 0 && 12 % 4 == 0        return 0
```

1 of 4



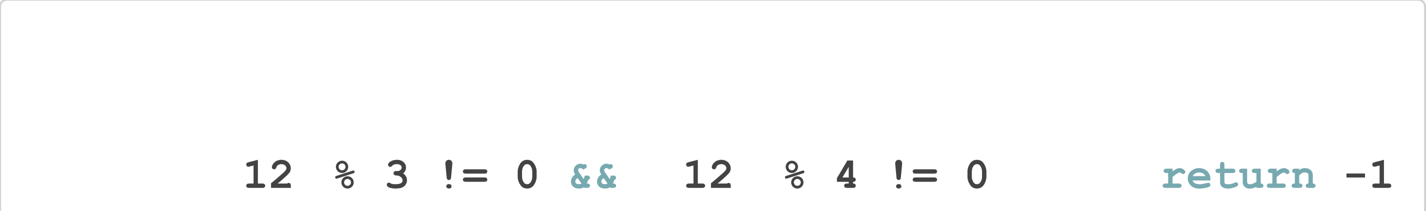
```
12 % 3 == 0 && 12 % 4 != 0        return 1
```

2 of 4



```
12 % 3 != 0 && 12 % 4 == 0        return 2
```

3 of 4



```
12 % 3 != 0 && 12 % 4 != 0        return -1
```

4 of 4

The next challenge will test you on how return an array of squares from a function.