Solution Review 1: Find The Factorial

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

Solution

```
fn test(n:i32) {
   let mut factorial = 1; // define a mutable variable factorial

   if n < 0 { // check if factorial is less than zero
      println!("0"); // print 0
   }
   else if n == 0 {  // check if factorial is equal to 0
      println!("1"); // print 1
   }
   else // go here if the above two conditions are false
   {
      for i in 1..n + 1{
         factorial = factorial * i
      }
      println!("{}", factorial); // print the factorial
   }
}
fn main(){
    print!("factorial (4) : ");
    test(4);
    print!("factorial (6) : ");
    test(6);
}
```

Explanation

- On **line 2** a mutable variable `factorial` is initialized to `1`.
- `if` **construct**
  - As factorials exist only for positive numbers, the condition `n < 0` on **line** 4 checks if the value of input number `n` is less than 0, it *prints* 0 on **line**

    4, checks if the value of input number `n` is less than 0 , it *prints 0* on **line 5**.

    if the `if` **condition fails** then `else if` is executed.

- `else if` **construct**
    - As the factorial of 0 is 1, the condition `n == 0` on **line** 7, checks if the value of input number `n` is equal to 0 then it *prints 1* on **line 8**.

      if the `else if` **condition fails** then `else` block is executed.

- `else` **construct**
    - A `for` **loop** is defined within the else block.
        - `for` loop definition

          On **line 12**, the `for` loop iterates from 1 to `n+1` times.

        - `for` loop body
            - Within each iteration, on **line 13** `factorial` is multiplied with `i` and the updated value is saved in `factorial`.
            - The loop iterates until the `i` is equal to `n + 1`.
    - The value of `factorial` is printed on **line 15**.

The following illustration explains the above code.

> Assume that the variable `n` is initialized with value 4 and passed to the function.

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4^4

1^1

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 { false
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

44

11

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {false
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

44

11

```
fn test(n:i32) {
   let mut factorial = 1;

   if n < 0 {
      println!("0");
   }
   else if n == 0 {false
      println!("1");
   }
   else
   {
      for i in 1..n + 1{
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

4^4

1^1

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {false
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

44

11

```
fn test(n:i32) {
   let mut factorial = 1;
                                                   4
   if n < 0 {
      println!("0");
                                                   1
   }
   else if n == 0 {false
      println!("1");                               1
   }
   else
   {
      for i in 1..n + 1{
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

1

1

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

```
factorial * i
1 * 1
=1
```

1

1

```
fn test(n:i32) {
   let mut factorial = 1;

   if n < 0 {
      println!("0");
   }
   else if n == 0 {
      println!("1");
   }
   else
   {
      for i in 1..n + 1{
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

4

1

2

```
fn test(n:i32) {
   let mut factorial = 1;

   if n < 0 {
      println!("0");
   }
   else if n == 0 {
      println!("1");
   }
   else
   {
      for i in 1..n + 1{
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

4

```
factorial * i
1 * 2
=2
```

2

2

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

2

3

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

```
factorial * i
2 * 3
=6
```

6

3

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

6

4

```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

4

```
factorial * i
6 * 4
=24
```

24

4

```
fn test(n:i32) {
   let mut factorial = 1;

   if n < 0 {
      println!("0");
   }
   else if n == 0 {
      println!("1");
   }
   else
   {
      for i in 1..n + 1{ i == 5 => Loop breaks
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

4

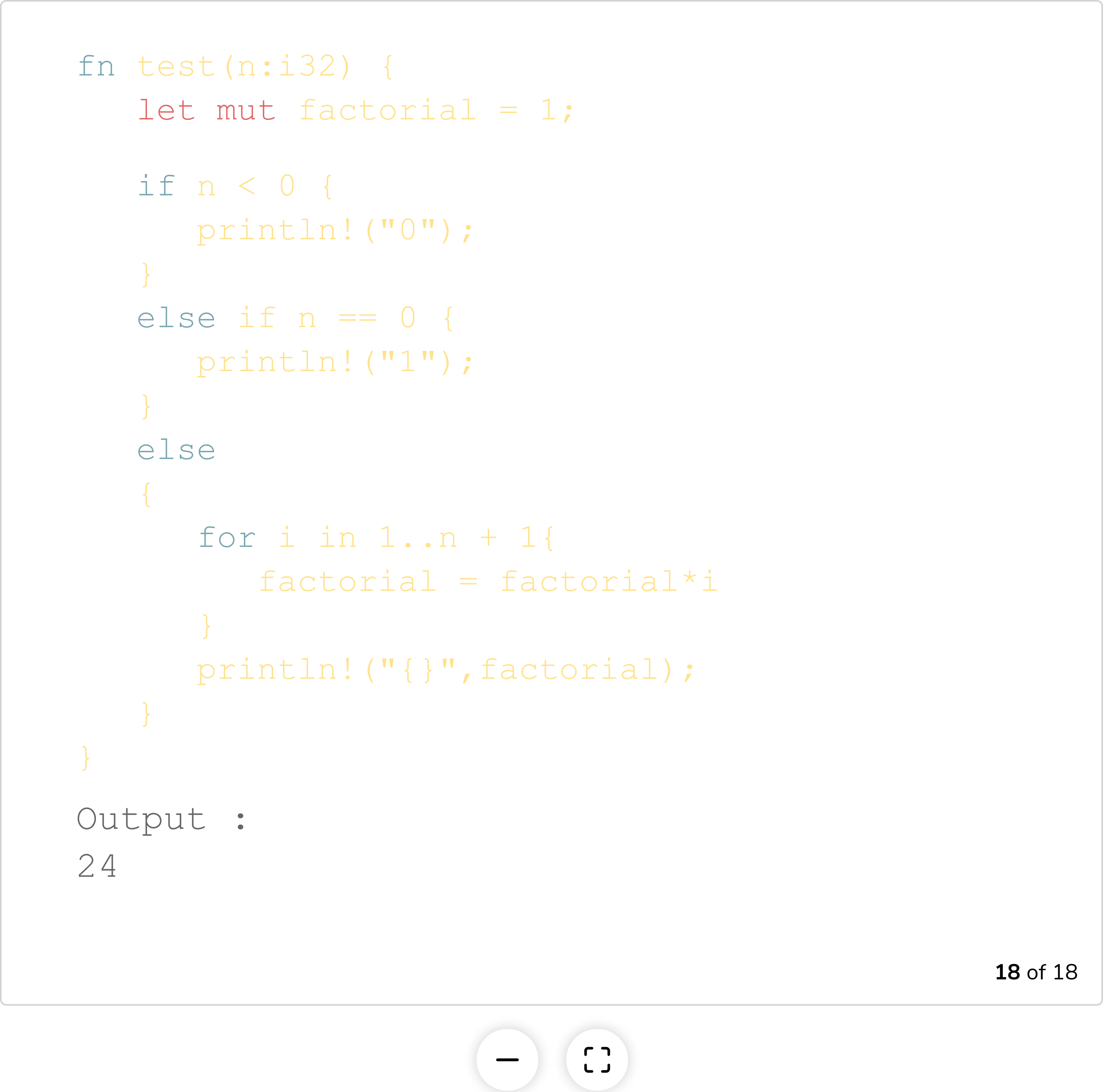
24

5

```
Output :
24
```

```
fn test(n:i32) {
   let mut factorial = 1;
                                                  4
   if n < 0 {
      println!("0");
   }
   else if n == 0 {
      println!("1");
   }
   else
   {
      for i in 1..n + 1{
         factorial = factorial*i
      }
      println!("{}",factorial);
   }
}
```

```
Output :
24
```



```
fn test(n:i32) {
    let mut factorial = 1;

    if n < 0 {
        println!("0");
    }
    else if n == 0 {
        println!("1");
    }
    else
    {
        for i in 1..n + 1{
            factorial = factorial*i
        }
        println!("{}",factorial);
    }
}
```

```
Output :
24
```