A Switch Expression Challenge

So, in the last challenge, we used a traditional switch statement, to translate a letter into NATO's keyword, that represented that letter.

In this next challenge, we're going to use the enhanced switch expression.

Let's look at these statements side by side again, but this time, we're going to make the enhanced switch an expression, by assigning it to a variable.

Traditional Switch Statement vs. Enhanced Switch Expression

Traditional Switch Statement used in a method, returning values

```java
switch (month) {
    case "JANUARY":
    case "FEBRUARY":
    case "MARCH":
        return "1st";
    case "APRIL":
    case "MAY":
    case "JUNE":
        return "2nd";
    case "JULY":
    case "AUGUST":
    case "SEPTEMBER":
        return "3rd";
    case "OCTOBER":
    case "NOVEMBER":
    case "DECEMBER":
        return "4th";
}

return "bad";
```

Enhanced Switch Expression

```java
return switch (month) {
    case "JANUARY", "FEBRUARY", "MARCH" -> { yield "1st"; }
    case "APRIL", "MAY", "JUNE" -> "2nd";
    case "JULY", "AUGUST", "SEPTEMBER" -> "3rd";
    case "OCTOBER", "NOVEMBER", "DECEMBER" -> "4th";
    default -> {
        String badResponse = month + " is bad";
        yield badResponse;
    }
};
```

Day of the Week Challenge

1. Create a method called printDayOfWeek, that takes an int parameter called day, but doesn't return any values.

   - Use the enhanced switch statement, to return the name of the day, based on the parameter passed to the switch statement, so that 0 will return "Sunday", 1 will return "Monday", and so on. Any number not between 0 and 6, should return "Invalid Day".
   - Use the enhanced switch statement as an expression, returning the result to a String named dayOfTheWeek.
   - Print both the day variable and the dayOfTheWeek variable.

Day of the Week Challenge

2. In the main method, call this method for the values 0 through 7.
3. Bonus: Create a second method called printWeekDay, that uses an if then else statement, instead of a switch, to produce the same output.