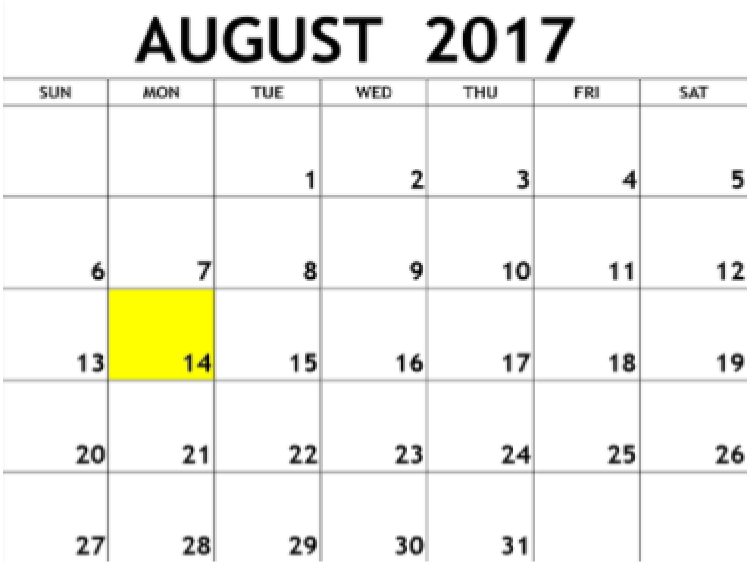
The Calendar class is an abstract class that provides methods for converting between a specific instant in time and a set of calendar fields such as YEAR, MONTH, DAY_OF_MONTH, HOUR, and so on, and for manipulating the calendar fields, such as getting the date of the next week.

You are given a date. You just need to write the method, , which returns the *day* on that date. To simplify your task, we have provided a portion of the code in the editor.

The method should return  as the day on that date.



AUGUST 2017

| SUN | MON | TUE | WED | THU | FRI | SAT |
|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 |
| 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 | 24 | 25 | 26 |
| 27 | 28 | 29 | 30 | 31 | | |

Complete the *findDay* function in the editor below.

*findDay* has the following parameters:

- *int:* month
- *int:* day
- *int:* year

**Returns**

- *string:* the day of the week in capital letters

**Input Format**

A single line of input containing the space separated month, day and year, respectively, in  format.

```
08 05 2015

WEDNESDAY
```

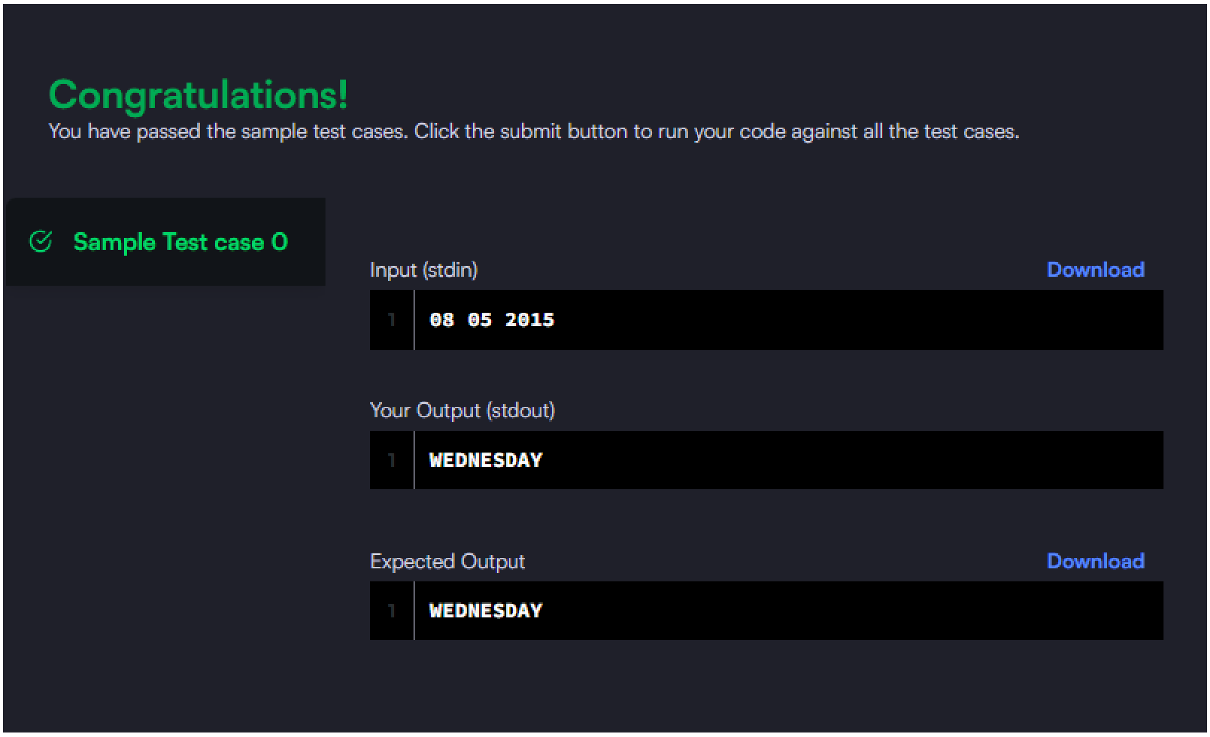
```
class Result {

    /*
     * Complete the 'findDay' function below.
     *
     * The function is expected to return a STRING.
     * The function accepts following parameters:
     *  1. INTEGER month
     *  2. INTEGER day
     *  3. INTEGER year
     */

    public static String findDay(int month, int day, int year) {
        { Calendar cal = Calendar.getInstance(); cal.set(year,
month - 1, day);
        // month is 0-based in Calendar
         String dayOfWeek =
cal.getDisplayName(Calendar.DAY_OF_WEEK, Calendar.LONG,
Locale.US);
         return dayOfWeek.toUpperCase();

    }
    }

}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
08 05 2015
```

Your Output (stdout)

```
WEDNESDAY
```

Expected Output

```
WEDNESDAY
```