

Homework 3

1. Calendar (continued)

In this exercise you have to extend the `Calendar` program that you wrote in Homework 2. Let us start by noting the following fact: 1.1.1900 was a Monday. Write a program that counts how many Sundays fell on the first day of the month during the 20th century (from 1.1.1901 to 31.12.2000, inclusive). In other words, you have to march through about 36,500 days and dates; if the current day happens to be the first day of the week as well as the first day of the month, you have to count it.

We recommend to start by writing a program that realizes a calendar. You can prove this capability by printing your calendar for some time range. For example, the following program prints the calendar (for testing purposes) between 1.1.1901 to 31.12.1902, inclusive (the printout below skips many days, and shows the output in two columns, to save space):

```
% java Calendar
1/1/1901                20/8/1901
2/1/1901                21/8/1901
3/1/1901                22/8/1901
4/1/1901                23/8/1901
5/1/1901                24/8/1901
6/1/1901 Sunday        25/8/1901 Sunday
7/1/1901                26/8/1901
...                    27/8/1901
1/8/1901                28/8/1901
2/8/1901                29/8/1901
3/8/1901                30/8/1901
4/8/1901 Sunday        31/8/1901
5/8/1901                1/9/1901 Sunday
6/8/1901                2/9/1901
7/8/1901                3/9/1901
8/8/1901                4/9/1901
9/8/1901                5/9/1901
10/8/1901               6/9/1901
11/8/1901 Sunday       7/9/1901
12/8/1901               8/9/1901 Sunday
13/8/1901               ...
14/8/1901               27/12/1902
15/8/1901               28/12/1902 Sunday
16/8/1901               29/12/1902
17/8/1901               30/12/1902
18/8/1901 Sunday       31/12/1902
19/8/1901               3 Sundays fell on the first of the month
```

Notice that 1.9.1901 is one of these special Sundays that we have to count. The other two special Sundays occurred during the dates marked above as "...".

You can check the correctness of your generated calendar by consulting this [website](#).

Implementation tips: We know that 1.1.1900 was a Monday. Basically, you have to run several cyclical counters: the day of the week (1, 2, ..., 7, 1, 2, ..., 7, 1, 2, ...), the day of the month (1, 2, ..., n_1 , 1, 2, ..., n_2 , 1, 2, ...) where n_1, n_2, \dots are 28, 29, 30, or 31, depending on which month and year we are in, the month of the year (1, 2, ..., 12, 1, 2, ..., 12, 1, 2, ...), and the year of the century (1901, 1902, ..., 2000).

Write your solution by completing the given `Calendar.java` class file.

The key function in this program is `advance`, which advances the counters mentioned above. We recommend starting by writing this function, incrementally: write code that advances one counter and prints the counter's values as you go along, for testing purpose. Then add another counter, print the values, etc.

2. Array operations

Write a function that computes and returns an array whose elements are the maxima of the respective elements of two given arrays. The size of the returned array is the size of the longer array. For example, if the two given arrays are `[3,4,1]` and `[2,6,8,5,1]`, the function returns `[3,6,8,5,1]`. Note that “extra elements” at the end of the longer array are copied as-is to the output array. Write your solution by completing the given `ArrayOps.java` class file.

3. String operations

Complete the given `StringOps` class. You have to write five functions. We recommend to implement and test them in the order in which they appear in the class. Tip: The `removeChar` function can be implemented in one statement.

Submission

Before submitting your solution, inspect your code and make sure that it is written according to our **Java Coding Style Guidelines**. Also, make sure that each program starts with the program header described in the **Homework Submission Guidelines**. Both documents can be found in the Moodle site, it is your responsibility to read them. Any deviations from these guidelines will result in points penalty. Submit the following five files only:

- `Calendar.java`
- `ArrayOps.java`
- `StringOps.java`

Deadline: Submit Homework 3 no later than November 11, 23:55. You are welcome to submit earlier.