© Manfred Kerber

School of Computer Science
University of Birmingham

Worksheet 2

MSc/ICY Software Workshop 1

Assessed Exercise: 10% of the continuous assessment mark.

Submission: Thursday 7 November 2013 12:00 noon

5% late submission penalty per 24 hours or parts thereof. No submission after 24 hours.
Tests and comments are mandatory. Submit your `.java` files via Canvas as a single zip file.

Exercise 1: (Basic, 30%) Write a program that checks for two strings consisting of characters `'a'-'z'` and `'A'-'Z'` whether they are equivalent irrespective of uppercase or lowercase letters.

Equivalent are:	Not equivalent are:
`"test"` and `"tesT"`	`"test"` and `"tea"`
`"test"` and `"TEST"`	`"test"` and `"tset"`
`"test"` and `"TesT"`	`"test"` and `"testS"`
`"TesT"` and `"tESt"`	`"test"` and `""`

Exercise 2: (Basic, 15%) Write a program that prints – without duplicates – the integers between 0 and a maximal number `maximum`, which are divisible by any of 2, 3, and 5, but not by any of 7 and 11 in increasing order. The numbers should be separated by spaces and only these numbers should be printed. In each line there should be exactly 20 numbers, except for the last which may contain fewer. Test your program for `int maximum = 300;`. Do not forget to comment what the loop does and why it terminates.

Exercise 3: (Medium, 15%) Bubble sort is a very simple sorting algorithm which swaps adjacent elements in an array which are out of order until the whole array is sorted. E.g.:

```
      [4, 3, 6, 1, 9, 2]
->    [3, 4, 6, 1, 9, 2]
->    [3, 4, 1, 6, 9, 2]
->    [3, 4, 1, 6, 2, 9]
->    [3, 1, 4, 6, 2, 9]
->    [3, 1, 4, 2, 6, 9]
->    [1, 3, 4, 2, 6, 9]
->    [1, 3, 2, 4, 6, 9]
->    [1, 2, 3, 4, 6, 9]
```

Give a Java implementation of bubble sort using loops.

Exercise 4: (Advanced, 20%) In Linux you can print an overview of the current month by the command `cal`.

Write a static method that generates a string with a month overview in the following format. Make use of variables `max` for the number of days in the month (31 in the example) and `first` for indicating the first day of the month, 0 for `Mo`, 1 for `Tu`, 2 for `Wed`, and so on (i.e., 2 in the example). Take care that the numbers 1 through 9 are aligned to the right as in the example.

```
Mo Tu We Th Fr Sa So
       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
```

Exercise 5: (Advanced, 20%) In Linux you can print an overview of a year by the command `cal 2013`, e.g. Write a corresponding program that generates a string with a year overview for any given year as in the example. Take as additional argument the first day in the year, e.g. 1 for 2013. Note you have to determine whether the year is a leap year, since then February has 29 days.

```
                            2013
      January               February               March
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6               1  2  3               1  2  3
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   4  5  6  7  8  9 10
14 15 16 17 18 19 20  11 12 13 14 15 16 17  11 12 13 14 15 16 17
21 22 23 24 25 26 27  18 19 20 21 22 23 24  18 19 20 21 22 23 24
28 29 30 31           25 26 27 28           25 26 27 28 29 30 31


       April                  May                   June
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7         1  2  3  4  5                  1  2
 8  9 10 11 12 13 14   6  7  8  9 10 11 12   3  4  5  6  7  8  9
15 16 17 18 19 20 21  13 14 15 16 17 18 19  10 11 12 13 14 15 16
22 23 24 25 26 27 28  20 21 22 23 24 25 26  17 18 19 20 21 22 23
29 30                 27 28 29 30 31        24 25 26 27 28 29 30


        July                 August              September
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
 1  2  3  4  5  6  7            1  2  3  4                     1
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   2  3  4  5  6  7  8
15 16 17 18 19 20 21  12 13 14 15 16 17 18   9 10 11 12 13 14 15
22 23 24 25 26 27 28  19 20 21 22 23 24 25  16 17 18 19 20 21 22
29 30 31              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30


      October               November              December
Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su  Mo Tu We Th Fr Sa Su
    1  2  3  4  5  6               1  2  3                     1
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   2  3  4  5  6  7  8
14 15 16 17 18 19 20  11 12 13 14 15 16 17   9 10 11 12 13 14 15
21 22 23 24 25 26 27  18 19 20 21 22 23 24  16 17 18 19 20 21 22
28 29 30 31           25 26 27 28 29 30     23 24 25 26 27 28 29
                                            30 31
```