Course 02101
Paul Fischer og Hjalte Wedel Vildhøj, DTU Compute

Indledende Programmering – Efterår 2016
Afleveringsopgave 1

Dato: 20. 09. 2016, **Afleveringsdato**: 7. 10. 2016, kl.23:59 på Campusnet

Dette er den første afleveringsopgave i kurset 02101. Husk at disse opgaver er *obligatoriske* og indgår i kursets samlede bedømmelse. Procenttallet for hver opgave indikerer den estimerede tid.

Læs den generele information om afleveringsopgaverne på Campusnet:
`Fildeling->Generel Information->VejledningAfleveringsopgaver.pdf` og
`Fildeling->Generel Information->Samarbejdspolitik.pdf`.
Husk, at programmerne skal køre i *CodeJudge*, hvor der også ligger et par testfiler.
Husk desuden: Ingen mellemrumsymboler før eller efter outputtet.

In general, make your programs robust, for example ensure that they can handle illegal inputs.

Afleveringsopgave 1.1: [**15%**] Check-digits are used to check whether a number is correctly entered by a user (for example a bank account number or CPR-number). By using check digits, many – but not all – errors can be found. The check digit is added to the original number such that the test can be easily performed. Many such algorithms are used, we describe one algorithm below. The number is given as a string and the check digit is already added at the end.
Let $x_{n-1}x_{n-2}\cdots x_2x_1x_0$, be an n-digit decimal number, that is $x_i \in \{0,1,2,3,4,5,6,7,8,9\}$, for all i. The algorithm for verifying a number replaces every digit with an odd index by twice the digit if that is less then 10. If twice the digit is 10 or larger then take the modulus with respect to 10 and add 1. That is, 2 is replaced 4, 7 by $5 = (14 \bmod 10)+1$. Then all the digits of the resulting number are summed up, i.e. the cross sum is computed. If that sum has remainder 0 modulo 10 the number is accepted, otherwise not. The digits with an even index are not changed.
Write a class `NumberCheck` which contains a method

```
public static boolean check(String number)
```

which returns true if the number is accepted and false otherwise. Use **exactly** the names specified above for the class and method.
For example, the number 3475 becomes 6455 after replacement of the odd-indexed digits and is accepted by the program because the cross sum of 6455 is 20 and $0 = 20 \bmod 10$. The number 41032 is rejected.
You may assume that the string `number` only contains decimal digits; you do not have to check that.
Hint: Do not try to do everything at the same time. Break the problem down into smaller tasks. For example: how to get the digits of the string the right order, how to turn a character into a numerical value, etc.

Slut på Opgave 1

Afleveringsopgave 1.2: [**30% & 20%**] Consider the $n \times n$ *grid of natural numbers*, that is, all pairs of numbers (x, y), where $x, y \in \{0, 1, \ldots, (n-1)\}$.
A particle is placed at a grid point with coordinates (x, y); the coordinates are chosen at random. It then starts to move around the grid. A move is done by changing either the x-coordinate or the y-coordinate. The length of the move is between 0 and s, the length is determined at random for every move according to uniform distribution. Also the direction – up, down, left, or right – is chosen randomly with equal probability. If the new coordinate is outside the grid, the particle is not moved. Such a process is called a *two-dimensional random walk*. The program terminates after t moves.

a) Write a program `RandomWalk` with a method

```
public static void runSimulation(int n, int t, int s)
```

which implements the simulation described above. The meaning of the parameters n, t and s is as explained above.

The program should write the parameters to the standard output in the first line in the following format, for example

```
n=5 t=2 s=2
```

Then it has to write the positions of the particle, one per line, including the start position in the following format:

```
[3;4]
```

Make sure that you use procedural decomposition for suitable sub-tasks. Consider whether it makes sense to use one or more of the library classes introduced in the course.

Here is a sample output:

```
n=5 t=4 s=2
[2;1]
[1;1]
[1;3]
[0;3]
[0;1]
```

b) Write a program `RandomWalk2` with a method

```
public static void runSimulation2(int n, int t, int s)
```

which has two particles. The meaning of the parameters is the same as above. Both particles are placed randomly on the grid. In a move the first particle moves as the one described in part a). The second particle moves differently; it is attracted by the first one and tries to reach it. The move of the second particle can be described as follows: First the distances between the x- and y-coordinates of the particles are computed. Then the second particle moves to make the larger of the two distances as small as possible, while not moving more than s grid points either horizontally or vertically. Both positions are printed after every move in the following format:

```
[1;2] [3;4]
```

When both particles are at the same position, the program prints the positions, the text "Crash", and stops.

Here is a sample output in case the particles meet:

```
n=5 t=8 s=2
[4;0] [1;3]
[3;0] [1;1]
[1;0] [1;0]
Crash
```

The programs which you upload do not have to to have a main method.

Slut på Opgave 2

Afleveringsopgave 1.3: [**35%**] Write a program ParkAutomat which controls a ticket machine for parking. Parking cost 30 dkk. per hour. The maximum parking time is 2 hours. The customer arrives at a randomly chosen time. Use the method randomTime() listed below and given on Campusnet.
The customer buys time by inserting coins (with values 20, 10, 5, 2, 1). The machine shows the end time for the parking. For illegal coins the text Illegal coin, try again. is shown.
The user can select “Buy” (B) or “Cancel” (C) with the obvious reactions. When one Types “T” the program writes Shutting down. and terminates immediately. If more money is inserted than the maximum parking time allows, the rest is returned and the option to insert coins is no longer shown. You can rely on, that only “B”, “C” and “T” is given as non-numerical input, however there strange coins might be used. When B is selected with no coins inserted, a ticket is printed nevertheless.
After a transaction is completed by buying or canceling, the time is set again using randomTime(). Do not use this method otherwise. Times are shown in the format HH:MM.
Use this code to generate the time for every new transaction. You program should have a main method which starts it.

```java
import java.text.DecimalFormat;
import java.util.Random;

    // put these two lines into the top of the class

    public static Random rand = new Random(53332274312L);
        public static DecimalFormat df2 = new DecimalFormat("00");

  public static String randomTime(){
              return df2.format(rand.nextInt(24))+":"+
                             df2.format(rand.nextInt(60));
        }
```

A sample dialog is shown below:

```
****************************
 The time is 13:18
 Parking time until 13:18
 Please insert Coins
 C - Cancel
 B - Buy
****************************
10
****************************
 The time is 13:18
 Parking time until 13:38
 Please insert Coins
 C - Cancel
 B - Buy
****************************
10
****************************
 The time is 13:18
 Parking time until 13:58
 Please insert Coins
 C - Cancel
 B - Buy
****************************
B
Your ticket is printed. Thank you
=== Transaction completed ===
****************************
 The time is 20:04
 Parking time until 20:04
 Please insert Coins
 C - Cancel
 B - Buy
****************************
20
****************************
 The time is 20:04
 Parking time until 20:44
 Please insert Coins
 C - Cancel
 B - Buy
****************************
20
****************************
 The time is 20:04
 Parking time until 21:24
 Please insert Coins
 C - Cancel
 B - Buy
****************************
2
****************************
 The time is 20:04
 Parking time until 21:28
 Please insert Coins
 C - Cancel
 B - Buy
***************************
20
Maximum Parking time reached. 2 dkk returned.
***************************
 The time is 20:04
 Parking time until 22:04
 C - Cancel
 B - Buy
***************************
5
Maximum Parking time reached.
***************************
 The time is 20:04
 Parking time until 22:04
 C - Cancel
 B - Buy
***************************
B
Your ticket is printed. Thank you
=== Transaction completed ===
***************************
 The time is 12:48
 Parking time until 12:48
 Please insert Coins
 C - Cancel
 B - Buy
***************************
20
***************************
 The time is 12:48
 Parking time until 13:28
 Please insert Coins
 C - Cancel
 B - Buy
***************************
C
Operation cancelled. Money returned 20 dkk.
=== Transaction completed ===
***************************
 The time is 20:15
 Parking time until 20:15
 Please insert Coins
 C - Cancel
 B - Buy
***************************
10
***************************
```

```
 The time is 20:15
 Parking time until 20:35
 Please insert Coins
 C - Cancel
 B - Buy
***************************
12
Illegal coin, try again.
***************************
 The time is 20:15
 Parking time until 20:35
 Please insert Coins
 C - Cancel
 B - Buy
***************************
C
Operation cancelled. Money returned 10 dkk.
=== Transaction completed ===
***************************
 The time is 02:02
 Parking time until 02:02
 Please insert Coins
 C - Cancel
 B - Buy
***************************
T
Shutting down.
```

Slut på Opgave 3