

Uge 4.

Exercise 1: Write a class `GradeConverter` which contains a method

```
public static int convert13to7(int grade)
```

which transforms a mark on the old 13-scale to one on the 7-step scale. The table below contains the rules.

7-trinsskalaen	13-skalaen
	12 13
	12 11
	10 10
	7 9
	7 8
	4 7
	02 6
	00 5
	00 03
-3	00

7-trinsskalaen ug.dk 13-skalaen

Make your program throw an `IllegalArgumentException` if the input is not a mark on the 13-scale.

_____ End of Exercise 1 _____

Exercise 2: Write a class `Min4Digits` which contains a method

```
public static String min4Digits(int n)
```

which takes an integer number n as input and returns it as a string with at least 4 digits, using leading zeros if necessary. For example 15 is shown as 0015. Test your implementation. If n is negative, a minus sign is put first in addition.

_____ End of Exercise 2 _____

Exercise 3: Write a program which ask the user to input a string. If the input contains the string "java" or the string "nova", then the program writes "yes" to the console and "no" otherwise. For the upload on CodeJugde, the program should not ask the user and only write the answer.

_____ End of Exercise 3 _____

Exercise 4: Write a method which checks whether a string represents a number (integer or real) or not. Test your implementation.

_____ End of Exercise 4 _____

Exercise 5: The method `System.currentTimeMillis()` returns the number of milliseconds since midnight, January 1, 1970 UTC (Coordinated Universal Time) as a `long`. Details can be found in the Java-API Documentation.

<http://docs.oracle.com/javase/7/docs/api/>

Write a method that converts this – or another value – to the number of years, month, days, hours, minutes and seconds and returns this as a nicely formatted string. What is the answer for 748430593656L?

_____ End of Exercise 5 _____

Exercise 6: Write a class `Sort4` which contains a method

```
public static void sort4(int a0, int a1, int a2, int a3)
```

The method should print the numbers `a0`, `a1`, `a2`, and `a3` sorted in increasing order, separated by blanks in single line.

Bonus question: (*) How many comparisons are needed to sort 5 integers? 7 integers?

_____ End of Exercise 6 _____

Exercise 7: Write a program which takes 3 numbers from `System.in`. If the first two numbers are equal up to the precision indicated in the third number, then the program prints 1 and otherwise 0. The third input is the amount of decimals to consider. For example, given the input

```
2.342
```

```
2.336
```

```
2
```

the program should print 1, because it compares 2.34 with 2.34 but for the input

```
2.3454
```

```
2.3456
```

```
3
```

the program should print 0, because it compares 2.345 with 2.346.

_____ End of Exercise 7 _____

Exercise 8: (*) Write a class `StringSum` containing the method

```
public static int sumOfIntegers(String text)
```

The method takes a string `text`, identifies all the integers in the string, and returns their sum.

For instance, the string `"a2.05h34-9fmq26"` contains the integers 2, 5 (05), 34, -9, and 26, so the method should return 58.

_____ End of Exercise 8 _____

Exercise 9:

Lav en klasse ved navn `AlphaOrder`.

a) Implementer følgende metoder:

```
public static String alphaFirst(String wordOne, String wordTwo)
```

Der returnerer det ord der kommer først, hvis de var sorteret alfabetisk rækkefølge.

```
public static String alphaSecond(String wordOne, String wordTwo)
```

Der returnerer det ord der kommer sidst, hvis de var sorteret i alfabetisk rækkefølge.

b) Implementer metoden med følgende signatur:

```
public static void alphabeticalOrder(String wordOne, String wordTwo,
String wordThree)
```

Der givet 3 strenge printer dem i alfabetisk rækkefølge.

b) Implementer foregående metode, men hvori der kun laves metodekald til `alphaSecond`, `alphaFirst` og printmetoder (`println/print`), det vil sige ingen `if`-statements.

_____ End of Exercise 9 _____

Exercise 10: Write a class `DistinctChecker` containing the methods:

a) `public static String distinctChars(String s)`

That given a `String s`, should return a `String` consisting of the distinct characters in `s`, in the order they first appear.

The distinct characters are the characters of the string without duplicates, eg. "abba" and "aaab" should both return "ab" and "programming" should return "progamin".

b) `public static boolean compareDistinctChars(String s1, String s2)`

That given two strings return true, if the distinct characters of the two strings are the same, disregarding order. It should return false otherwise.

Eg. "abba" and "baba" should return true, as the distinct characters of both strings are 'a' and 'b'. For the input "abc" and "bad", on the other hand, it should return false.

End of Exercise 10

Exercise 11: [*] Write a class `CaesarCipher` which encrypts messages after Caesar's method: Given the 26-letter alphabet `A, ..., Z`, a text consisting of letters from this alphabet, and a natural number $k \in \{0, \dots, 25\}$. The coding is done by cyclically left-shifting the alphabet by one, e.g., for $k = 2$ we get

original	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
ciphertext	C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

Then the plaintext `INTRODUCTIONTOPROGRAMMING` is encoded as ciphertext `KPVTQFWEVKQPVRTQITCOOKPI`. The decoding is done by reversing the shift.

Your program has to have methods

```
public static String encode(String text, int shift)
public static String decode(String text, int shift)
```

which perform the encoding and decoding and return the respective result. In case the input is illegal, the string `***ERROR***` is returned.

Hint: You might want to use the ASCII number of the letters.

End of Exercise 11