

## 1 Palindrome Recognition (5 points)

Write code using the available functionality of `StringBuilder` and `String`, that recognizes if an input string is a *palindrome*, e.g. if it's the same "sentence" no matter if read forward or backwards. The case of characters, spaces and sentence markers should be ignored!

Write unit tests testing your, e.g. using the following strings:

```
"Erika feuert nur untreue Fakire."
```

```
"Ida war im Atlas, Abdul lud Basalt am Irawadi."
```

## 2 Decompose and Print a URL (10 points)

The following text is part of the official URL specification:

### Scheme

Within the URL of an object, the first element is the name of the scheme, separated from the rest of the object by a colon. The rest of the URL follows the colon in a format depending on the scheme.

### Internet protocol parts

Those schemes which refer to internet protocols mostly have a common syntax for the rest of the object name. This starts with a double slash `"/"` to indicate its presence, and continues until the following slash `"/"`. Within that section are

- **An optional user name**, if required (as it is with a few FTP servers). The password, if present, follows the user name, separated from it by a colon; the user name and optional password are followed by a commercial at sign `"@"`. The use of user name and passwords which are public is discouraged.
- **The internet domain name** of the host in RFC1037 format (or, optionally and less advisably, the IP address as a set of four decimal digits)
- **The port number**, if it is not the default number for the protocol, is given in decimal notation after a colon.
- **Path** The rest of the locator is known as the "path". It may define details of how the client should communicate with the server, including information to be passed

transparently to the server without any processing by the client. The path is interpreted in a manner dependent on the scheme being used. Generally, the reserved slash "/" character (ASCII 2F hex) denotes a level in a hierarchical structure, the higher level part to the left of the slash.

Write a class `Url` where the constructor takes a URL string and decomposes it in the previously described parts and stores them in appropriate fields. For the RFC1037 format you can assume that this is a sequence of names that contain only letters and are separated by dots ("."). for the **Path**, you can assume that each part consists of a string of arbitrary characters except "/".

Use the classes and methods for regular expressions in Java. If the string is not compatible with the URL specification, a `MalformedURLException` should be thrown.

In addition, implement a `toString()` method that returns the original URL string from which the URL object was created. Consider the recommendations given in the lecture for creating such a string in multiple steps.

Write unit tests to verify your code works.