Exercise Sheet #2

How to submit your solutions
Here some advice how to make communication more efficient.

(a) In order to get the points for solving a problem you need to submit your fully **commented and compilable solution** in time.

(b) Therefore send all relevant files **named by your name and the problem** it solves, e. g. `SmithJoe_problem2a_additionalInfo.xyz`.

(c) **Never** use a space-character in a file name!

(d) The solution should be handed in through the OLAT system (detailed instructions will be added to the course webpage).

(e) Grades are given on a scale of 0 to 20 points. You are free to solve any of the given exercises, but 20 is still the maximal grade.

(f) The **deadline for submission** is Monday of the week in which the sheet is discussed.

Problem 1 (*Odd Or Even*) 5 Pts

Write a program that receives a number as input and checks whether it is even or odd:

(a) Use `cin, cout` to receive an integer number as input and output a message declaring it is odd/even.

(b) Write an if statement checking if the number is odd or ever. You can use the modulus operator to create a boolean expression that is True only for even numbers.

(c) **Optional:** Write the program as short as you can, using the least number of characters (or lines) possible.

Problem 2 (*GIF Manipulation*) 5 Pts

Use it to manipulate a GIF:

(a) Go to `https://giphy.com/` and download an animated GIF by right-clicking on it and choosing 'save image as...' make sure you save it as a .gif file.

(b) Use `convert <your GIF> frames%04d.png` to convert the GIF to a sequence of images.

(c) Using `convert -delay 10 -loop 1 frames*.png animated.gif`, you can convert this back to an animated GIF.

(d) By changing the parameters of `convert`, change the speed of the GIF and make it loop several times.

(e) **Optional:** Wrap these commands into a convenient shell script that takes the animated GIF file as a command line argument.

Problem 3 (*Gnuplot*) 5 Pts

Create a gnuplot script following the instructions below:

(a) Set up the terminal and output commands to create a pdf file with color as an output. **Hint:** For output terminals see: `http://www.gnuplotting.org/output-terminals/`.

(b) Set the x-axis range to (0, 10) and the xlabel to `x`, and the y-label to `f(x)`.

(c) Plot the two functions $e^{-1/x}$ and e^{-x} in the same plot, using different line widths, line types and line colors. Give appropriate titles to each function. **Hint:** For plotting functions see: `http://www.gnuplotting.org/plotting-functions/`.

(d) Create the output file by calling `gnuplot <scriptname.gnu>`.

The next two items are optional: Create a gif by combining plots of e^{-a*x} for values of a ranging from 1 to 10:

(e) Write a shell script that contains a for loop over 1...10, calling the gnuplot script with the the iterated variable as a value for a. Furthermore, you should modify your script such that the saved image is not overwritten by appending a running index to your file names, e.g. `out-<index>.png`.

(f) Use the script from problem 2 to generate an animated gif from your image files.

Problem 4 (*Graphviz*) 5 Pts

A great tool for creating graph plots is Graphviz: `https://graphviz.org/`. Read about the DOT language and how to use it to write scripts that produce visual graphs.

Consider the following script:

```
# dot -Teps test.dot -o test.eps
graph G {
0 -- 1
1 -- 2
2 -- 0
        }
```

Run it on your computer to generate a graph with 3 connected nodes. Now make changes to the script:

(a) Add more nodes, and connect them to existing nodes.

(b) Make the graph directional by adding arrows to the lines (vertices) connecting the nodes.

(c) Change the colors of some vertices.

(d) Change the shape of some nodes to a box shape.

Problem 5 (*Shell Script, Advanced*) 10 Pts

Write a shell script called `mySpamCreator.sh` that sends an email every minute to an email address given to the script as an argument. You can choose the text you are sending with the email, but also attach a time stamp. In order not to crash any mail account make sure that the programm terminates after having sent maximally 5 mails.

Hint: A quick overview of linux commands is given here:
`https://gist.github.com/LeCoupa/122b12050f5fb267e75f`.

For sending emails from command line you can use the programm `mutt`. The typical syntax looks like:

```
echo "my message!" | mutt -s "my subject" -- email@adress.to
```

Next: Modify the program such that it sends an email at a given point in time. Try to send the 'spam' to several adresses, which you read from a file.