Assignment 1: Aug 24-30 (Git and bash intro)

Deadline: Sept 4

Important: All solutions of this assignment should be saved in a subdirectory named `week1` of your private git repository.

1.1 Your first git repository (2 pts)

Install git (download). Clone your private assignment repository. This repository should have the form UiO-INF3331/INF3331-{}, where {} is replaced by your name. First, create a `week1` folder in your git repository. Then add a textfile in that directory, add it to the repository and commit it. The textfile should be named Readme.txt and contain the words "Hello world". Push your first commit to github.

Name of scriptfile: Readme.txt

1.2 Resolve git conflicts (0 pts, but recommended)

Pulling from a git repository fails if collaborators pushed conflicting changes since the last pull. In this case the merge of conflicting changes need to be performed manually. Try to simulate such a scenario and fix the resulting merge conflict:

1) Add, and push a new file "gitconflict.txt" to your git repository.
2) Clone your git repository again into a different directory.
3) In both repository directories commit conflicting changes by adding different content to gitconflict.txt.
4) Attempt to push the changes of both repositories to github. The second push will fail and you will need to resolve the conflict manually.

1.3 Bash calculator (2 pts)

Write a script that takes as input a string containing a mathematical expression, evaluates the expression and prints the result to the screen. Example of use:

```
./calc.sh "5+5"
```

Should give output like

```
5+5 = 10.
```

Name of scriptfile: calc.sh

1.4 Compress large files (4 pts)

Write a script that takes as input a directory (path) name and a minimum file size in kilobytes. The script shall search the given directory tree, find all files of at least the given file size, and compress each of these files with gzip. Write a corresponding decompression script that takes a directory (path) name, finds all .gz files in that directory tree and decompresses them.

Both scripts should print out an error message if the wrong number of arguments is passed in, and a useful helper message if called with a "-h" or "–help" argument.

Hint: Use `find` and the `-size +200k` argument to find the files.

Name of scriptfile: compress.sh and decompress.sh

1.5 Eternal loop interrupted with Ctrl-C (2 pts)

Write a script that takes no input arguments, and repeatedly writes the current date and time (using the `date` command). The loop should be eternal, but when interrupted by Ctrl-C the script should print "Bye bye" to the screen before exiting. Hint: Use the bash line `trap control_c SIGINT` to capture the Ctrl-C interruption.

Name of scriptfile: clock.sh

Total points: 10