Assignment 2 (bash)

All solutions should be stored in a directory called `assignment2` in your private repository.

2.1: Calculator

Make a bash script which can read a bunch of numbers from the command line and print their sum, product, maximum or minimum. Should be exectuable as `./calc.sh ?  1 2 3 4 5 ...` where ? is S for sum, P for product, M for maximum and m for minimum. Examples:

```
$ ./calc.sh S 2 5
7

$ ./calc.sh P 4 2 3
24

$ ./calc.sh M 5 3 3 2
5

$ ./calc.sh m 5 3 3 2
2
```

Name of file: calc.sh
Points: 3

2.2: Clock

Make a simple clock by creating a bash script which, when run, prints the current time every second. To avoid making the terminal cluttered, have it clear the terminal before printing so only the current time is visible. Also give the script a command line option so that if run with the flag `--AMPM`, time is printed in the AM/PM-format (6:30 PM), while if the flag is not present, time is printed in the 24h format (18:30).

Name of file: clock.sh
Points: 2

2.3: Personal assistant

Make a script which looks up the departure times of the subways from Forskningsparken for you. These can be found on ruter.no. The script should print the next departure time for each of the six subways departing from Forskningsparken if called with no arguments. If called with the flags `--E` or `--W`, instead print only eastbound or westbound[1] subways respectively. Also add the

[1] If there is no clear east/west direction on the stop, just pick whichever you like - i.e. choose one direction and have `--E` show the subways leaving in that direction, and make `--W`

option for a command line argument to be given to specify another station to show subways from, and add support for Blindern as well as whichever subway station is closest to your home.

Clarifications and some pointers: by “specify another station” you do *not* need to make a script which can take an arbitrary station name and fetch the next departure times from that station - what is meant is that you make your script able to show the next departure times from Blindern, Forskningsparken or a third station you like, and make the script able to switch between showing the three by a commandline argument. I.e. `./subway.sh Blindern` might show Blindern times, while `./subway.sh Nydalen` might show Nydalen times. This makes the problem a bit easier because you can then hardcode the subway line names and search for the appropriate ones in the HTML.

If you need some pointers to ways of searching a string for some substring and cutting out parts of it based on the position of that substring (useful for parsing HTML), one way is parameter expansion. A brief introduction can be found at `http://wiki.bash-hackers.org/syntax/pe#substring_removal`. For example, a way of grabbing the “innermost” parentheses from a string with lots of singly nested parentheses is the following:

```
long_string=”(apples(oranges(pears)bananas)coconuts)”
long_string=${long_string##*(}
long_string=${long_string%%)*}
echo $long_string                    #pears
```

Name of file: subway.sh
Points: 5

show the subways leaving in the other direction.