Homework I : Introductory Linux and Shell Scripts

Guidelines for Homework I : Please read carefully!

1. Homework I is due Thursday, 11/10 by 17:00.
2. Homework will be submitted via email to `ustunel@metu.edu.tr`.
3. **VERY IMPORTANT!** The subject line of your email MUST read Phys343 Homework – nothing more, nothing less.
4. **VERY IMPORTANT!** Homework turned in on Thursday, 11/10 between 17:00-24:00 will only receive 50% of the full credit. Homework turned in later than 24:00 on the due date will NOT be accepted.
5. Please include the **a separate file** for each of the questions as attachment to your emails. There are four questions in this homework each asking you to write one shell script, so I'm expecting four shell scripts from you. Before sending me your scripts make sure that they work.

Goal of the assignment : In the lecture and lab sessions held this week, you learnt how to use shell commands and briefly saw how to develop shell scripts. In this homework you are going to practice in more detail how to write shell scripts with the help of lecture and lab notes that are available online and the `man` pages. I strongly suggest you read the notes before you start the homework.
Files to download : Go to the web site of the class and download the files for Homework I. They are called sun1.txt, sun2.txt, morning-sun.txt, evening-sun.txt and out.si-wire.

Question I : Write a shell script that takes in as argument a filename and displays its modification date and time in this way :

```
hande@p439a:~$ sh mod-date.sh sun1.txt
The file sun1.txt was modified on 2007-10-03 at 12:56.
hande@p439a:~$ sh mod-date.sh sun2.txt
The file sun2.txt was modified on 2007-10-01 at 2:15.
```

Hints :

1. Remember one of the options for `ls`, which we discussed in class.
2. Remember the pipe.

* Bonus for 2 points — you don't have to complete this part, it's optional : How would you obtain the following output instead?

```
hande@p439a:~$ sh mod-date.sh sun1.txt
The file sun1.txt was modified on 03/10/2007 at 12:56.
hande@p439a:~$ sh mod-date.sh sun2.txt
The file sun2.txt was modified on 01/10/2007 at 2:15.
```

Hint : You can use intermediate shell variables.

Question II : Now modify your shell script from Question I in such a way that it lists the modification dates for files whose names contain a given pattern.

```
hande@p439a:~$ sh mod-date-pattern.sh sun
The file sun1.txt was modified on 2007-10-01 at 01:26.
The file sun2.txt was modified on 2007-10-01 at 19:10.
The file morning-sun.txt was modified on 2007-10-01 at 02:53.
The file evening-sun.txt was modified on 2007-10-01 at 02:55.
```

Hints :

1. Construct a `for` loop.
2. Use the * sign.

Question III : For this question, we will be working with a real output file produced by a real scientific code. Embedded in a lot of junk, the essential information contained in the file is energies and atomic positions that are updated and written at every step. Suppose you are given the output file out and you would like to process it in two ways : extract the energies (very similar to what's done in lab) and extract the latest atomic positions. Write a shell script that takes in the name of the file and the number of atoms (114 in this case) as input and outputs the following :

```
hande@p439a:~$ sh extract.sh out 114
Energy : -1090.29131927 Ry.
The final positions are :
5.038926827 5.039056911 -0.234948454
20.622666005 20.622574227 -0.234908412
20.680870012 10.334081861 0.199678284
                  ⋮
```

Hints:

1. Make sure that your output follows the desired format, e.g. no `Si` at the beginning of each line of the atomic positions, the energy written with its units etc.
2. To display "`Energy :`" and the final energy value on the same line you have to use `echo` and one of its options.
3. The relevant keyword you can use to search for the energy values is `!`.
4. The relevant keyword you can use to search for the atomic positions is `ATOMIC_POSITIONS`. You need one of the options of `grep` to display `n` lines after the pattern being searched. Remember that you have to display the last 114 atomic positions displayed in the output with the `ATOMIC_POSITIONS` keyword.

Question IV : Usually, shell scripts are not well-equipped to do arithmetics but rely on the output from other programmes. But simple operations with integers can be performed using the shell command `expr`.

Write a script that takes in an integer as input and

1. if the integer is greater than 3, multiply it by two,
2. if it is smaller than 3, multiply it by three.
3. if it is equal to 3, display an error message.

So the execution should look like

```
hande@p439a:~$ sh multiply.sh 4
result = 8
hande@p439a:~$ sh multiply.sh 3
You've entered 3. Enter a different number.
hande@p439a:~$ sh multiply.sh 1
result = 3
```

Hints:

1. You can use the `if` construct.
2. Read the lecture notes for examples of the usage of `expr`.