Computer practicum I 2022/2023

Assignment 2 - Linux and Bash

Write down the commands you would use for the following tasks. If needed, submit the Bash script as well.

When writing solutions to the tasks, ***use a font that writes all letters with the same width*** *(e.g. courier new or consolas).*
Submit homework solutions in ***one PDF document*** *named* ***name_surname.pdf*** *via e-classroom by the deadline specified in the classroom.* Additionally, where needed submit Bash scripts as well. *Also, don't forget to* ***write your name in the document itself****!*
The answers to each question should be clearly marked.

Question 1)

(30 points)

Loti got a new job, but she is a bit lost, so she needs your help. Her boss who does not know her way around the PC, gave Loti access to her computer to do the following tasks:

Tasks:

1. Create a new folder named `photos`
2. Find all `jpg` and `jpeg` and `png` files on her computer and copy them to a folder `photos`
3. Delete all the `png` files in the new folder `photos` (created in task 1)
4. Compress the folder `photos` to a file `photos_personal.gz` and remove the original folder
5. Save history of commands to a file `name_surname_task1.txt`

For submission, submit all commands used for the above exercise. And the file `name_surname_task1.txt`.

Question 2)

(30 points)

Text processing, analysis and interaction with the user is very common in Linux. With this in mind, write an executable interactive script for analysis of a *`csv`* file about user expenses and income, that can be found here:

https://raw.githubusercontent.com/Marina225/CP1/main/expenses_people.csv

To complete this exercise, follow the following tasks.

Tasks:

1. Create a shell script (`name_surname_task2.sh`) that can be executed and read by anyone and edited only by the owner.
2. Sort the file by the *Income* column in ascending order and save to a new file `~/ordered_expenses.txt`.
3. Ask for the username they are interested in and extract the expenses and income for a selected user.
   - If the username requested is not found in the file, inform them "*`the name cannot be found`*" and exit the programme with an error message
4. Ask if they are interested to know how often a certain pattern occurs in the file
   a. If "`yes`" is selected: ask them for a pattern they are interested in and count the number of its occurrences in the file
   b. If "`no`" is selected: say goodbye and exit the program

For submission, submit the script `name_surname_task2.sh`.

Question 3)

(30 points)

Peter has been given an assignment by his Professor to strip the links and the text name from the HTML pages.

A html link is of the form,

```
<a href="http://vocabulary.famnit.upr.si">Vocabulary Training</a>
```

Where `a` is the tag and `href` is an attribute which holds the link Peter is interested in. The text name is "`Vocabulary Training`". Peter notices that the text name can sometimes be hidden within multiple tags

```
<a href="http://vocabulary.famnit.upr.si"><h1><b>Vocabulary Training</b></h1></a>
```

Here, the text name is hidden inside the tags `h1` and `b`.

Help Peter in listing all the links and the text name of the links. Implement a bash script with regex for this task.

Input Format
The first line contains the number of lines in the fragment (`N`). This is followed by N lines from a valid HTML document or fragment.

Output Format
If there are M links in the document, display each of them in a new line. The link and the text name must be separated by a "," (comma) with no spaces between them.

Strip out any extra spaces at the start and end position of both the link and the text name before printing.

```
link-1,text name-1

link-2,text name-2

....

link-n,text name-M
```

Sample Input:1

```
<p><a href="http://www.quackit.com/html/tutorial/html_links.cfm">Example Link</a></p>

<div class="more-info"><a href="http://www.quackit.com/html/examples/html_links_examples.cfm">More Link Examples...</a></div>
```

Sample Output:1

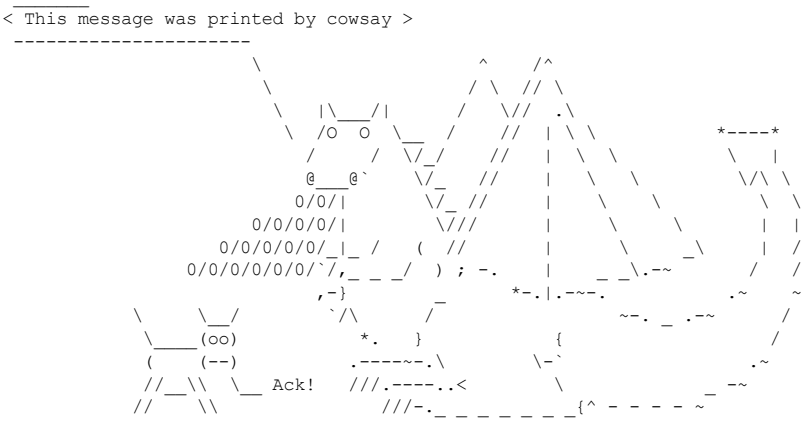
```
http://www.quackit.com/html/tutorial/html_links.cfm,Example Link

http://www.quackit.com/html/examples/html_links_examples.cfm,More Link Examples...
```

Question 4)

(10 points)

“Cowsay” is a configurable talking/thinking cow, originally written in Perl by Tony Monroe. It's a package that prints different figures into the terminal as a combination of many symbols and letters.



Peter, a computer science student at FAMNIT wanted to install this package on his Linux instance and found out he doesn’t have superuser privileges on the FAMNIT student server to

do so. So he decided to compile the binary on another device and execute the binary on his FAMNIT instance. So he compiled and committed the binary into the following repository.
https://github.com/nuwanarti/cowsay.git

Friends of Peter also would like to use this package on their instances as well. Your task is to help Peter to write a script that sets this package on their instances. Submit the script that you created.
The script should do the following:

1) Check for prerequisite packages "`git`"
    a) Should display an error if "`git`" is not installed on the computer
2) Should clone the git repo from the terminal (hint: `git clone https://github.com/nuwanarti/cowsay.git`)
3) Clone the repo if it's available on "Desktop", if not should display an appropriate error.
4) Set execution permission on the binaries, "`cowsay`", "`cowthink`"
5) Create two alias in `~/.bashrc` for the two binaries described in step 4.
6) Users should be able to use "`cowsay`" or "`cowthink`" commands from any directory they would like. `(like a regular command, hint: source, alias)`
7) Display a "`Thank You`" message at the end of the script as below
8) Display instructions on using the application.

```
 _____________________________
< Thankyou for installing cowsay >
 -----------------------------
\                             .       .
 \                           / `.   .' "
  \                 .---.  <    > <    >  .---.
   \                |    \  \ - ~ ~ - /  /    |
          _____          ..-~             ~-..-~
         |     |   \~~~\.'                    `./~~~/
        ---------   \__/                        \__/
       .'  O    \     /               /       \  "
      (_____,    `._.'               |         }  \/~~~/
       `-----.         /       }     |        /    \__/
             `-.       |       /      |       /      `. ,~~|
                 ~-.__|      /_ - ~ ^|      /- _      `..-'
                      |     /         |     /     ~-.     `-. _  _  _
                      |_____|         |_____|         ~ - . _ _ _ _ _>
```

For submission, submit the created script.