**Operating Systems**

**OPTIONAL ASSIGNMENT**

**BASH SCRIPTS**



**BACHELOR'S DEGREE IN COMPUTER SCIENCE AND ENGINEERING**

**Year 2018/2019**

David Rico Menéndez 100384036

Javier Ruiz Plaza 100383436

# Exercise 1

## 1.a

The script uses a while loop to go through all the arguments, inside the while it will check if the argument exists inside the /etc/passwd file, if it exists it will take the requested parameter from that line, else an error message will display. After an arguments have been processed it will shift the arguments one to the left making the second argument the first and so on.

## 1.b

Same as in the first part of the exercise.

# Exercise 2

## 2.a

First translates all assigns all non alphanumeric characters to new line characters taking as input the first argument and then sort sorts the result alphanumerically after that uniq grabs only the unique occurrences and puts an index in front of them indicating the number of occurrences of that string then is sorted again this time according to the string numerical value and put them in reverse order lastly head will print only the 10 first lines unless it's expecified in the second argument the number of lines to print

## 2.b

In order to extend the script so it verifies that 2 arguments are passed we only need to add an if conditional at the start of the script checking that ‘$#’ (number of arguments passed) is equal to 2, otherwise print an usage message

# 

# Exercise 3

First we check that the arguments are correct, in this case make sure the numbers are not negative and the second one is bigger than the first. Then in a while loop that will go through all the numbers in the interval will check if the factor of the number being processed is the same as the original number in that case the number is prime and will be printed.

# Exercise 4

First check that the output is one and it's an existing directory then the script will move to the directory if the checks pass. Then a for loop will look each file with format .jpg. Then the size of the jpg will be checked in case it's bigger than 1MiB using ImageMagick the jpg will be resized to 720 pixels in width, ImageMagick will automatically convert the height proportionally.

# Exercise 5

## 5.a

The file size of <exercise5.sh> was 1,1k at the beginning of the process, containing some code lines and random letters ending in “==”

After the execution of repackage, we can observe that the file size is 74,2 MB. Also we realize that, editing the file, the amount of random letters has expanded so far.

## 5.b

As mentioned before the program is designed to repack itself. Exit 0 is necessary in order to end the program and finish when all the data is loaded. The code 2>&1 we’ll say that is the most important part of the program, since is the command that is in charge of redirecting the Standard output and error (file descriptor 2) to the file chosen in the descriptor 1

## 5.d

This archive is going to perform several steps. THe first one will be to create a temporary file, that later will be use, and we are going to refer to it as the variable FILE, and the original file will be DEST. The first out of 3 steps will be to extract DEST into FILE. Then, we will resize DEST using truncate, that uses sed command to modify the content of it. Then we just add into the file all the directories that we are willing to use.