

Lab 11

Bonus Lab

Read carefully the instructions before performing!!!

Submission in pairs : 04/03/25 at 8:30

1. In this question we would like to implement a new bash command called `isPalindrome`. Do this according to the following steps:

1.1 Write a function called **`reverseNum`** that accepts a number and prints it in reverse order. For example, for the number 1234 the function will print 4321.

Note: Do not use `rev` or `awk` commands.

1.2 Use 1.1 and write a function called **`isPali`** which receives a number, checks if it is a palindrome and prints an appropriate message.

Note: A palindrome is a number whose reading from right to left and from left to right is the same. For example, 1364631 or 7117.

1.3 Complete the task so that a new bash command called **`isPalindrome`** is created. This command can be run and get an answer as to whether the given number is a palindrome.

For example, as a result of using the command
`>isPalindrome 12321`

we will see the message: The number is palindrome.

Hint: Use **`alias`**.

2. Write a script that accepts the name of a directory and displays the number of executable files in each one of its subdirectories recursively.

3. At Tel-Chen College it was decided to update the list of birthdays for all students. The college has a file named **`birthday_list`** that contains the birthdays of all the students in the college sorted by first names.

Each line in the file appears in the following format:

`first_name second_name birthday`

Below is an example of such a file:

```
Peppe Gerevsky 10/10/1990
Rina Dodina 23/02/1999
Taima Hanna 17/03/1995
```

3.1 Write a script called **`add`** that adds a new line to the file. The script accepts as parameters a first name, last name and date of birth (there is no

need to check the correctness of the parameters) and adds the appropriate line to the file.

The format of the command is:

add <first_name> <second_name> <birthday>

At the result, the birthday_list file will also contain the new line and will be sorted by first name. If the name already exists in the file, assume that the purpose of the command is to update the date (in case it was wrong). In this case, the line with the updated date will appear in the file, for this name. It can be assumed that there are no two people with the same name in the college.

3.2 Using 3.1, write a script called **add_birthdays** that adds several (one or more) lines to the file. The format of the command is:

add_birthdays <first_name> <second_name> <birthday> [<first_name> <second_name> <birthday>]

An appropriate message must be printed in case the number of parameters passed to the script is invalid (no parameters were given at all or the number of parameters is not divisible by 3).

3.3. Write a script called **search_birthday** that searches for all people born in a certain month.

The format of the command is:

search_birthday <birthday_month>

The script will output to stdout the names of all the people born in the requested month or alternatively it will inform that there are no people born in the requested month in the list.

Notes:

- It can be assumed that the input file is correct.
- The file shown above is only an example of the format of the group of input files.
- Do not use temporary files.
- Do not use awk to solve the question.

4. Definition: A person's BMI is a integer number calculated as follows:

$$\frac{Weight * 10000}{Height * Height}$$

where

Weight = the person's weight

Height = the person's height.

The LaBriut Health Insurance Fund is interested to find overweight customers in order to tailor a unique weight loss program to them. To help the health insurance fund, you must write a Bash script that will print to the standard output data regarding the BMI of the insurance fund's customers according to different cut-offs as described below.

In the directory where the program runs, there are files with the format **ageYY.data** and they contain all the members of the HMO who are **YY** years old. Each file contains lines of the form:

Client_id	Clie_name	Client_Height	Client_Weight
------------------	------------------	----------------------	----------------------

For example,

if in the file age43.data appears a line of the form

310000057	Moshelsraeli	172	82
-----------	--------------	-----	----

this means that a 43-year-old customer with ID number 310000057 is 172 cm

tall and weighs 82 kg. This client's BMI is equal to $\frac{82 * 10000}{172 * 172} = 27$

The program must operate in two different ways:

1). By passing a single numerical parameter, for example 23, then the program output will show the identity number and **age** of each client with a BMI greater than or equal to 23.

2). By passing 2 numerical parameters, 25 and 27, then the program output will show the identity number and **age** of each customer with a BMI between 25 and 27.

Remarks:

1. The identity number of the same customer must not be printed twice.
2. Do not use temporary files.
3. Any given input file has one or more lines.
4. The numbers 23, 25, 27 are given only as an example. You should work with general parameters.
5. Do not use awk to solve the question.

Submission:

Create one compressed file named `Linux_Lab11_id1_id2.tar`, where id1 and id2 must be changed to the id numbers of the two partners.

This file will include the files

Submit **only** the file `Linux_Lab11_id1_id2.tar` in Moodle and only by **one** of the partners.

Appeals:

Can be submitted by e-mail to Elad mail@eladhuttner.net within a week from the date of publication of the grades.