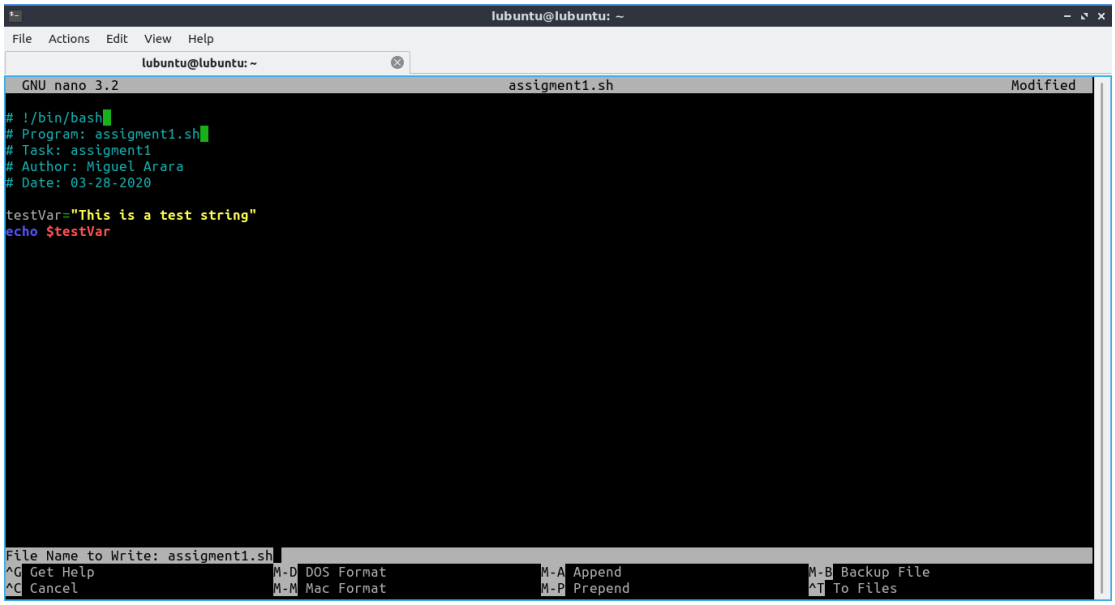
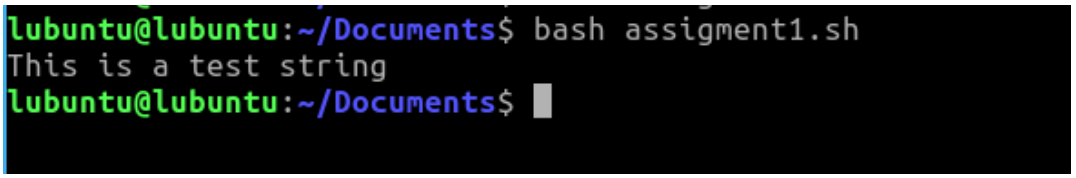
Scripting Languages / Assigments for the 1st lab

Assigment 1

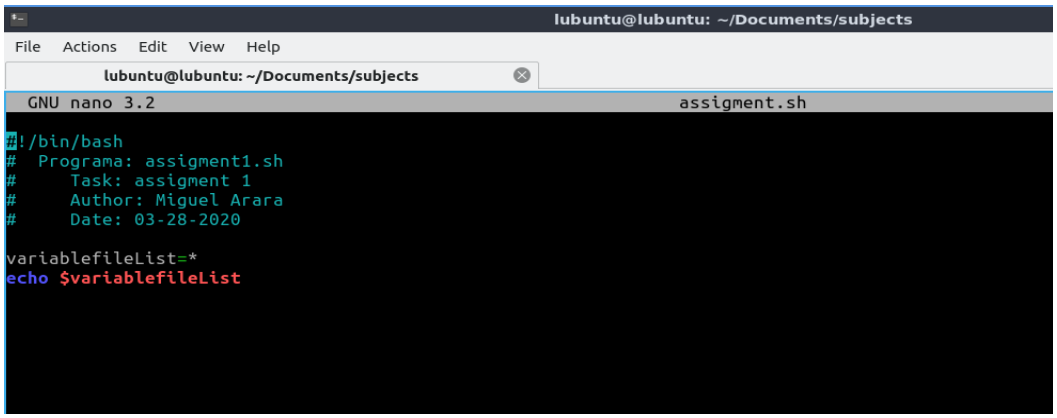
- Created the assigment1.sh



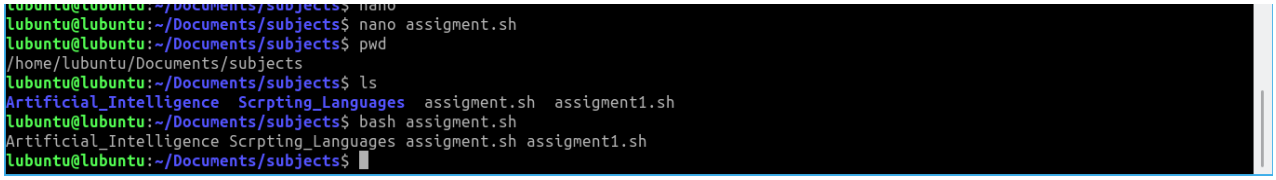
- Display the value of the variabletestVar.



- Store the list of all filenames contained in the current directory into the variablefileList.Use the filename expansion. Display the result

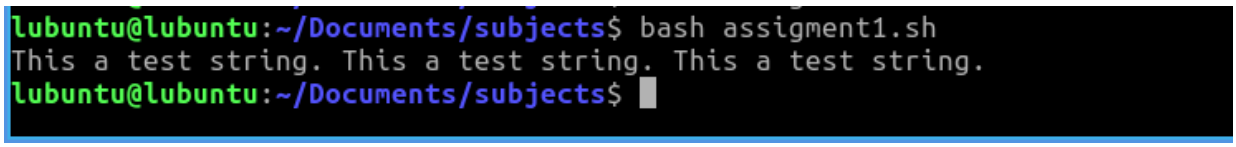


Executed the script, print all files in the current director.

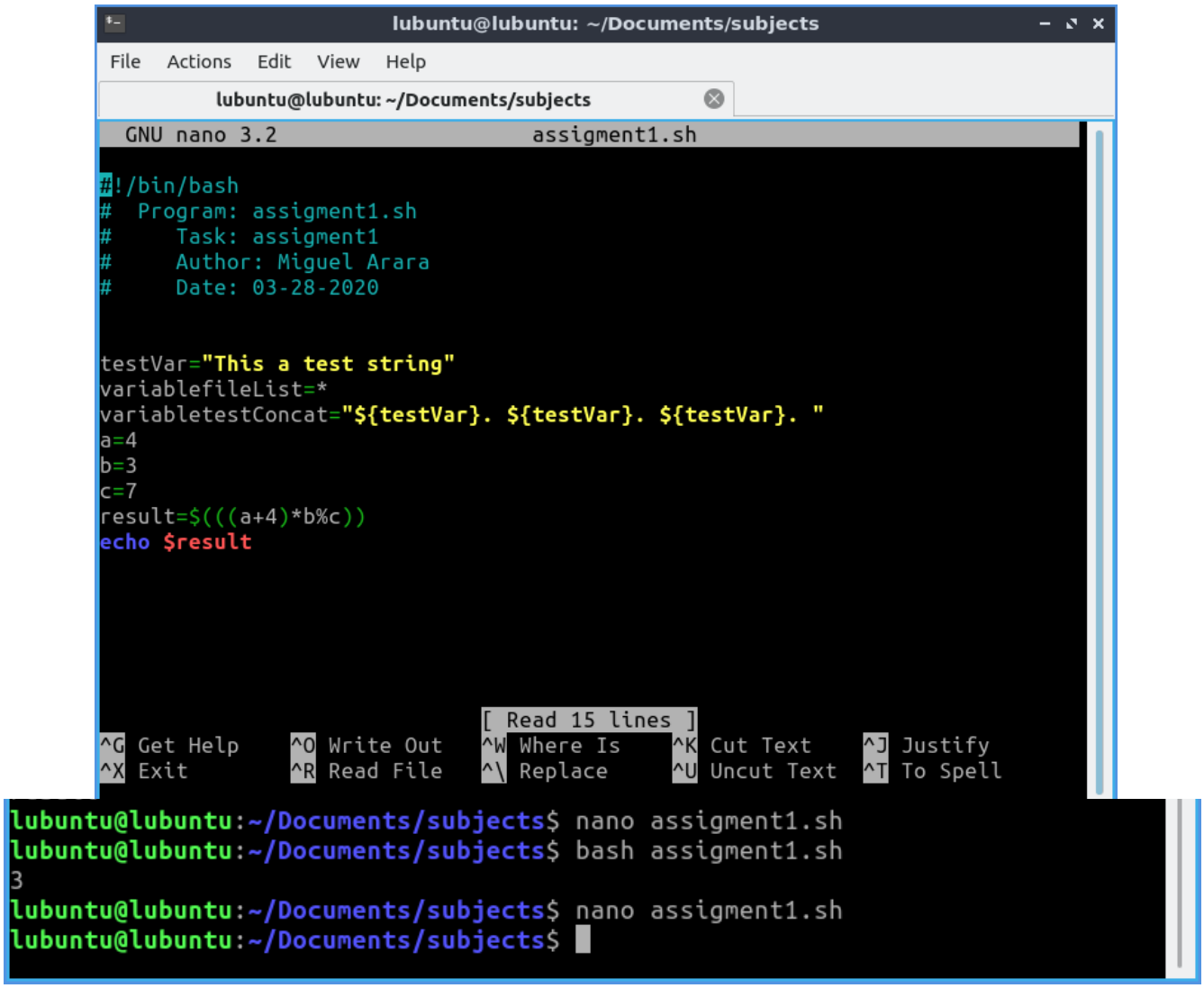


- Concatenate the sentence stored in the variabletestVarthree times. Each sentence shouldend by an added full stop and a blank. Assign the result to the variabletestConcat.

The result of variabletestConcat>



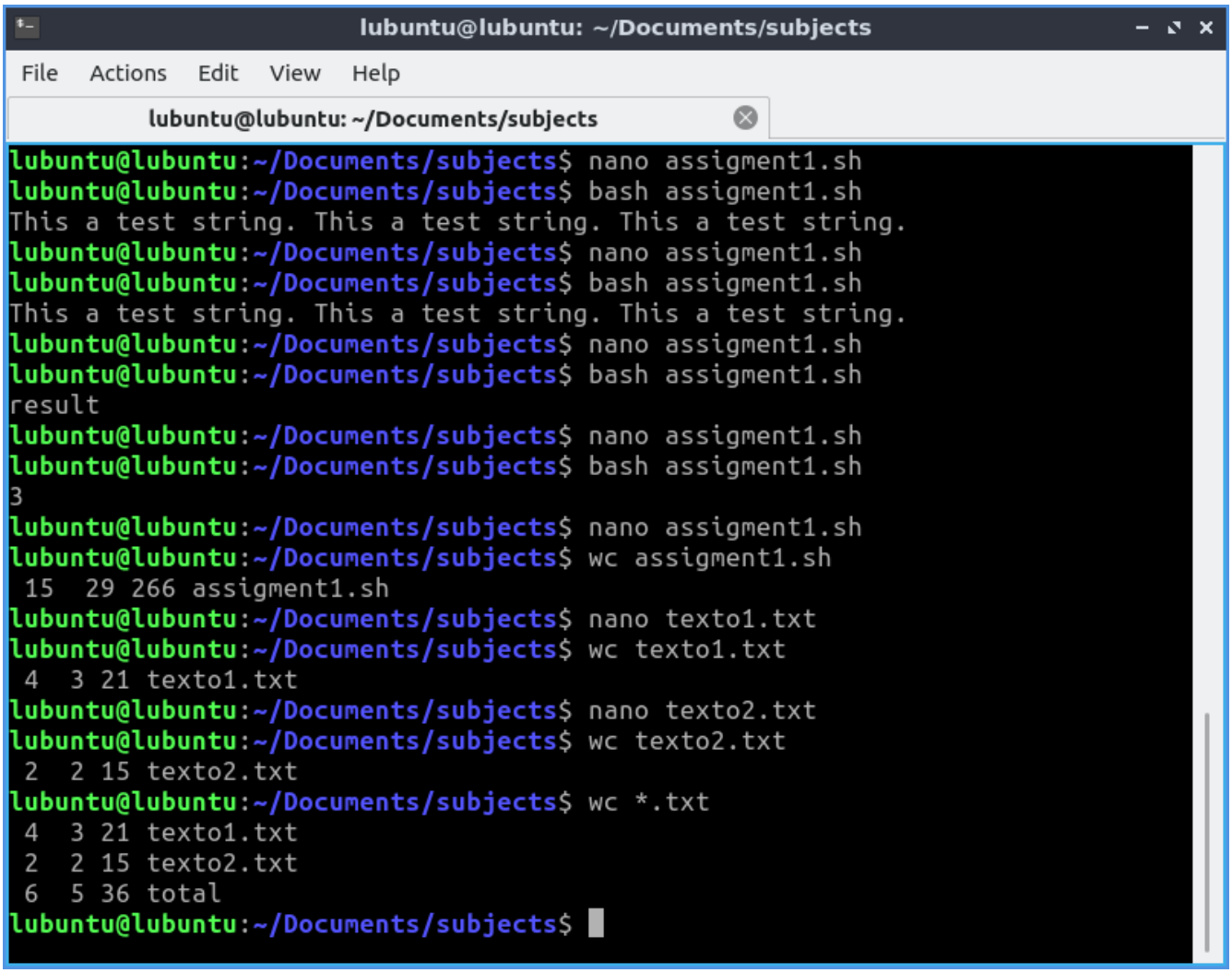
- Set the variable a to value 4, variable b to 3, variable c to 7. Set variable d to the value obtained by evaluating the expression(a+ 4)∗b%c. Use the arithmetic expansion.



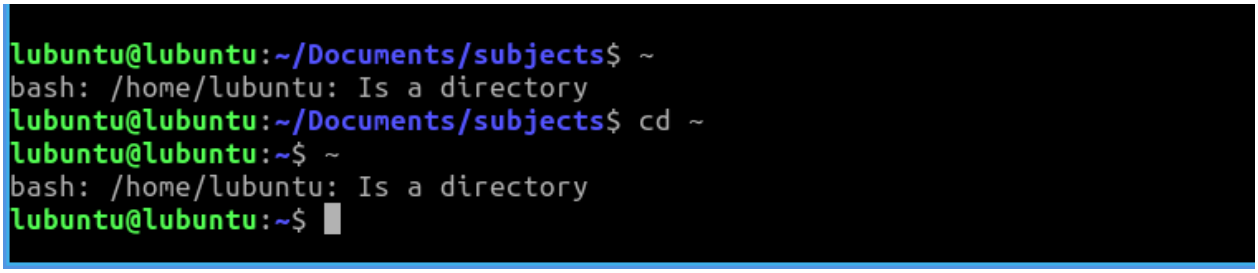
The result is 3.

- Set the variable number_of_words to the number of words in all.txt files in the current directory. Use the command substitution and the wc command.

I have two .txt files in the current directory with 4 and 2 words in each one. In total 6.



- Print the content of your home directory by using the tilde expansion

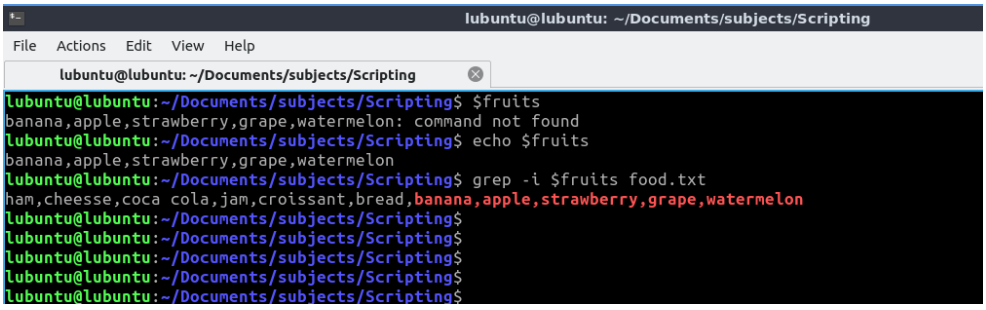


- Use a cut command to display (only) the information about user name, home directory andlogin shell of each user. The information is to be read from the/etc/passwd file.

Assigment 2

Command grep, regular expressions, command find. Loops.

- Write a grep command to find all lines in the file food.txt containing names of fruits(banana, apple, strawberry, grape, watermelon). The search should be case-insensitive.
    - I have created fruits array, and I have used in grep.



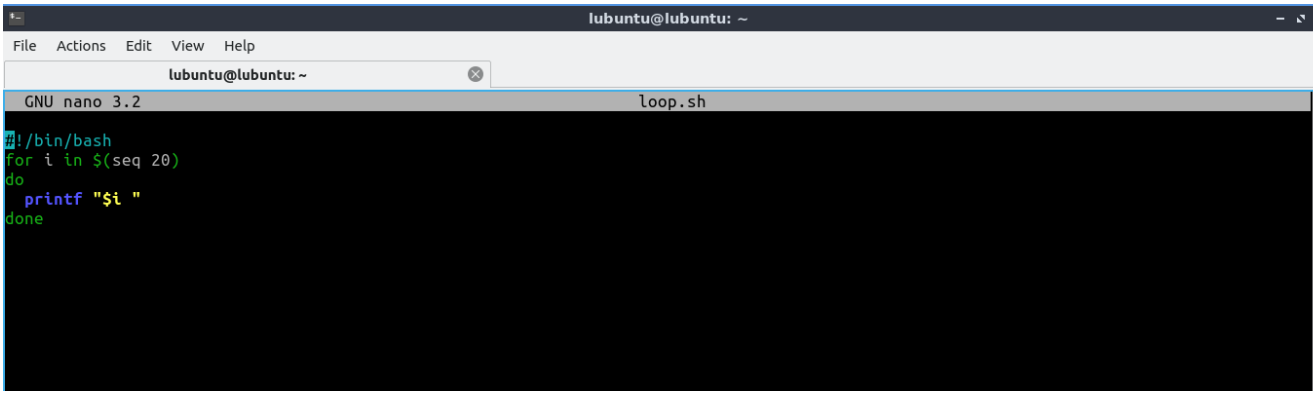
- Modify the command so that it prints only the lines not containing any of the listed words. Redirect the output to a file named non-fruit.txt.
    - With grep -v

- Write a grep command to find all files in the directory~/projects/and all subdirectories containing a code consisting of three capital letters and six digits (e.g. ABC123456) and display ll corresponding lines.

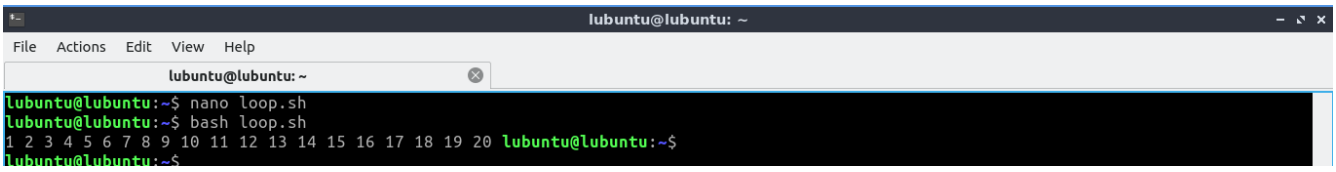
- Write a command which prints the names and details of all files in the current directory which were last modified before 7 to 14 days.

```
lubuntu@lubuntu:/etc$ find . -mtime -7 | find . -mtime -14
.
./.pwd.lock
./NetworkManager/system-connections
./NetworkManager/system-connections/Erasrooms.nmconnection
./apt
./apt/apt.conf.d
./apt/apt.conf.d/50unattended-upgrades
./apt/apt.conf.d/00aptitude
./apt/sources.list
./casper.conf
find: ‘./cups/ssl’: Permission denied
./cups
find: ‘./cups/ssl’: Permission denied
./cups/subscriptions.conf
./default/locale
./fstab
./group
```

- Write a one-line for loop which prints integer numbers between 1 and 20. Use the sequence generating expression or seq command.

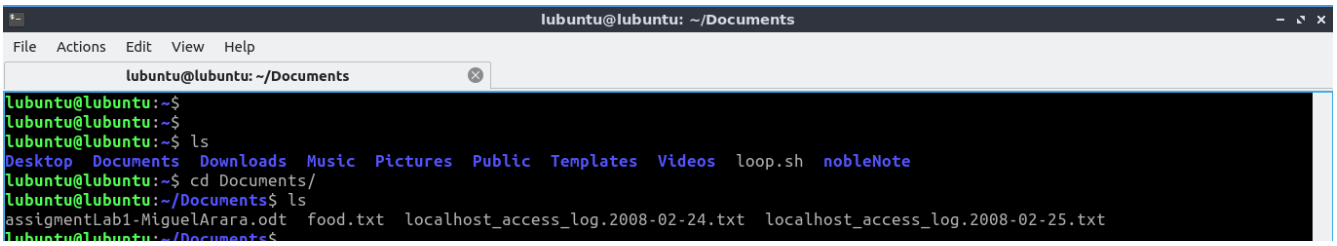


This is the result.

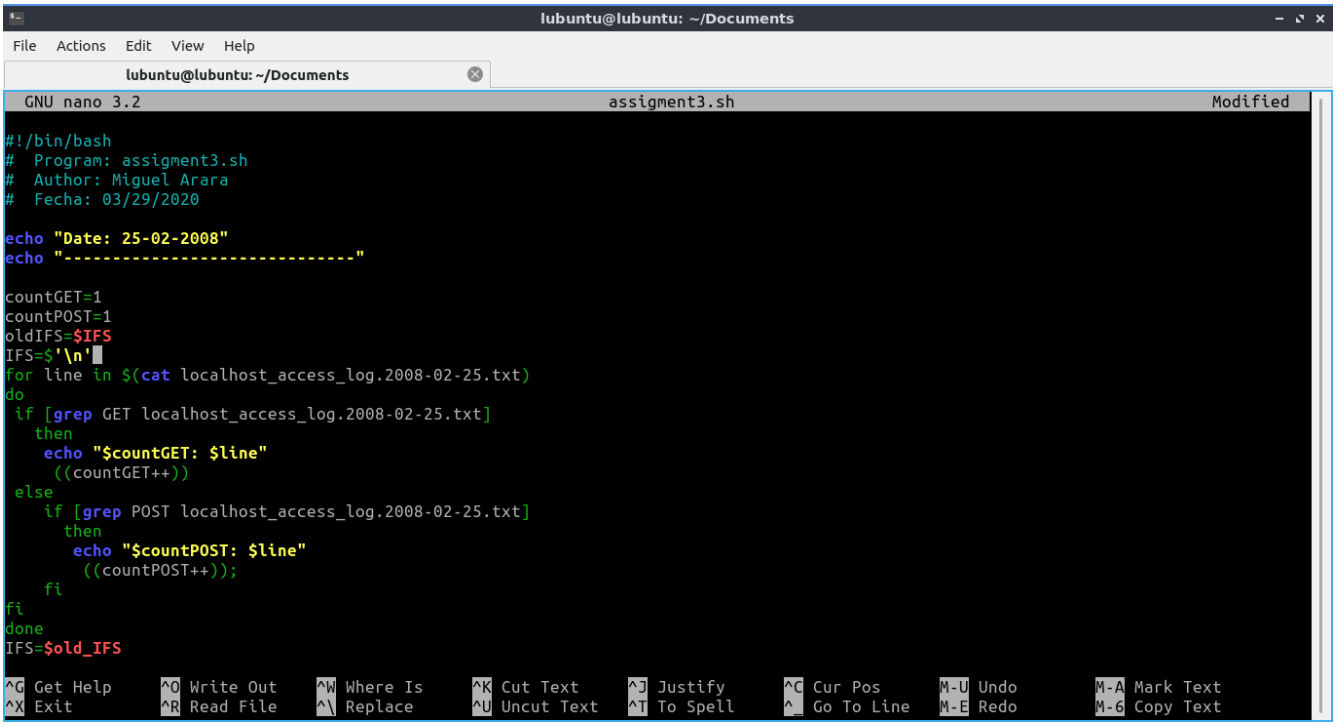


Assigment 3

I download both files .txt in that directory:



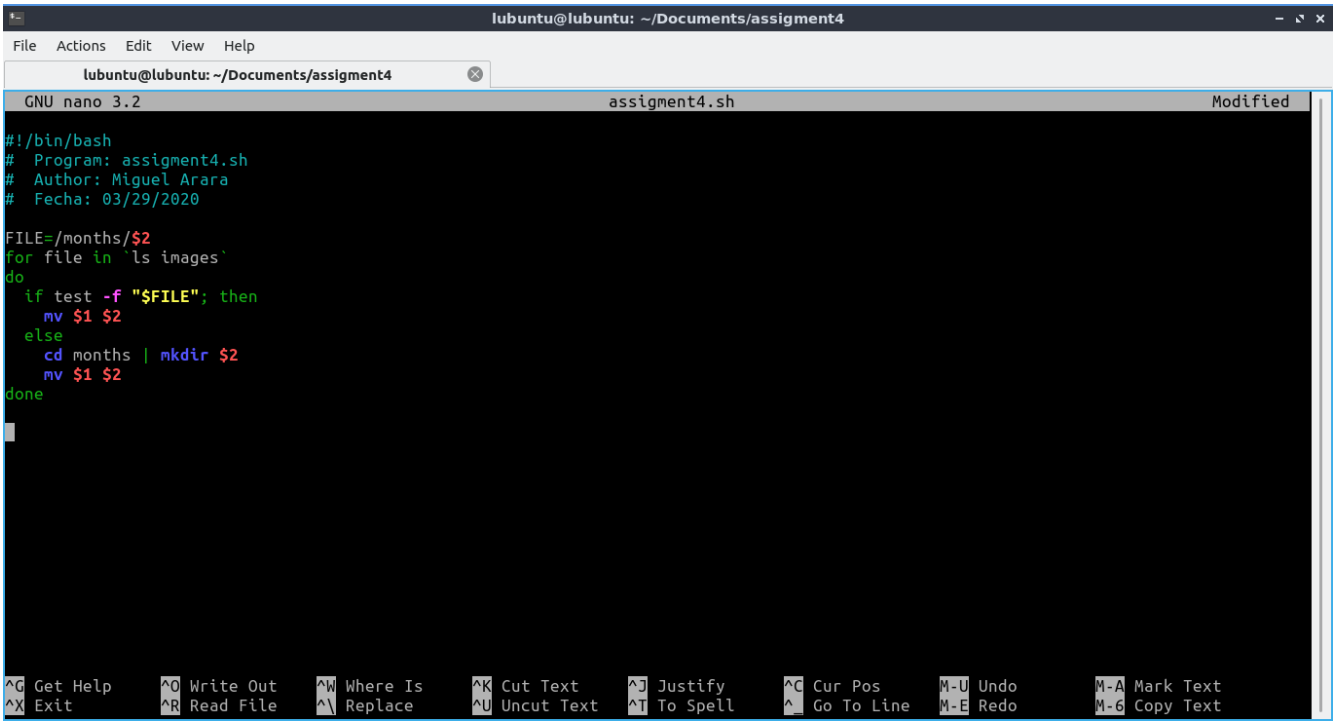
- Write a bash script that will analyze all log files created in February (according to the filename) and print the statistics of the requested methods. For each file the script should print the date and then for each action (the requested method and web page) print the number of occurrences in that file. The output list should be sorted according to the number of action occurrences, in descending order.



Assigment 4

- Suppose that one subdirectory inside the current directory contains photographs from our camera. The time of modification corresponds to the time of taking the photo. Write a script to move the photographs into folders corresponding to the months (the folder name should be in the format yyyy-mm). The script should accept two command line arguments: name of the source directory containing the images, and the name of the destination directory (where month directories are to be created). The script should check if the needed directory exists, create it, and move each image into the corresponding directory.
  - For example $1=images $2=2009-03

    > assigment4.sh images 2009-03



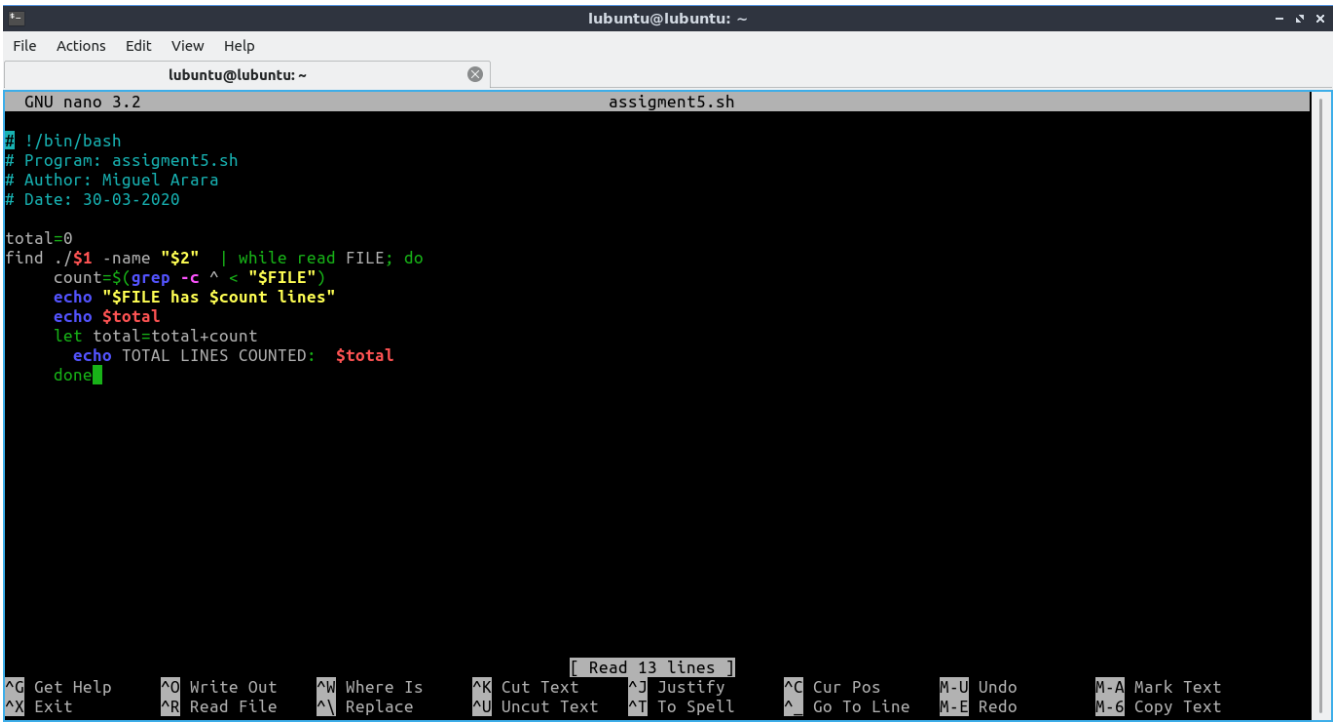
Assignment 5

- Write a script which will count the lines in all files in the directory given as the first command line argument and its subdirectories. The second command line argument should specify the filename pattern (e.g.'*.c'). The script should print the given command line arguments and then the total number of lines
    - The script should accept two command line arguments: name of the source directory containing the images, and the name of the destination director

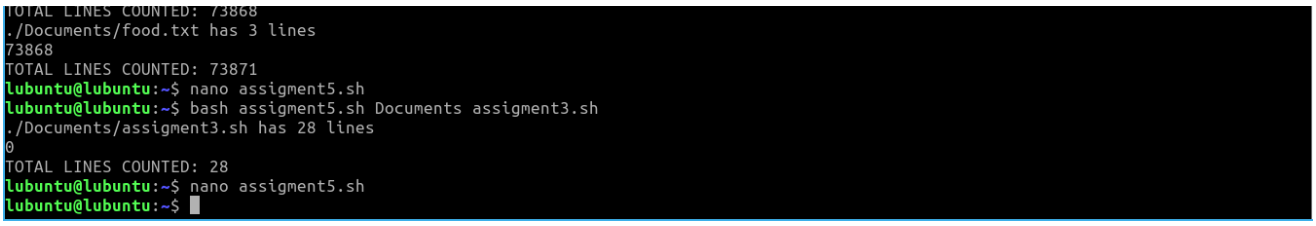
This program counts all lines of all files.

If we want put both line arguments only I have to do a couple of changes:

- Use $1 and $2 for use the line arguments
- Simply, I put the parameters:



OUTPUT: ($1=Documents, $2=assigment3.sh) → Result: 28 lines



Assignment 6

Write a script which will make a backup copy of the files given as command line arguments. The last command line argument should specify the destination directory. The script should check if the destination directory exists and create it otherwise. For each filename in the list the script should check the existence and readability. In the case of unsuccessful check, a message should be printed. The script should print a message about the directory creation and report the number of successfully copied files.

Note:In order to access the last command line argument, you can use indirect reference to$#variable:${!#}.

Here is an illustration of the script execution:

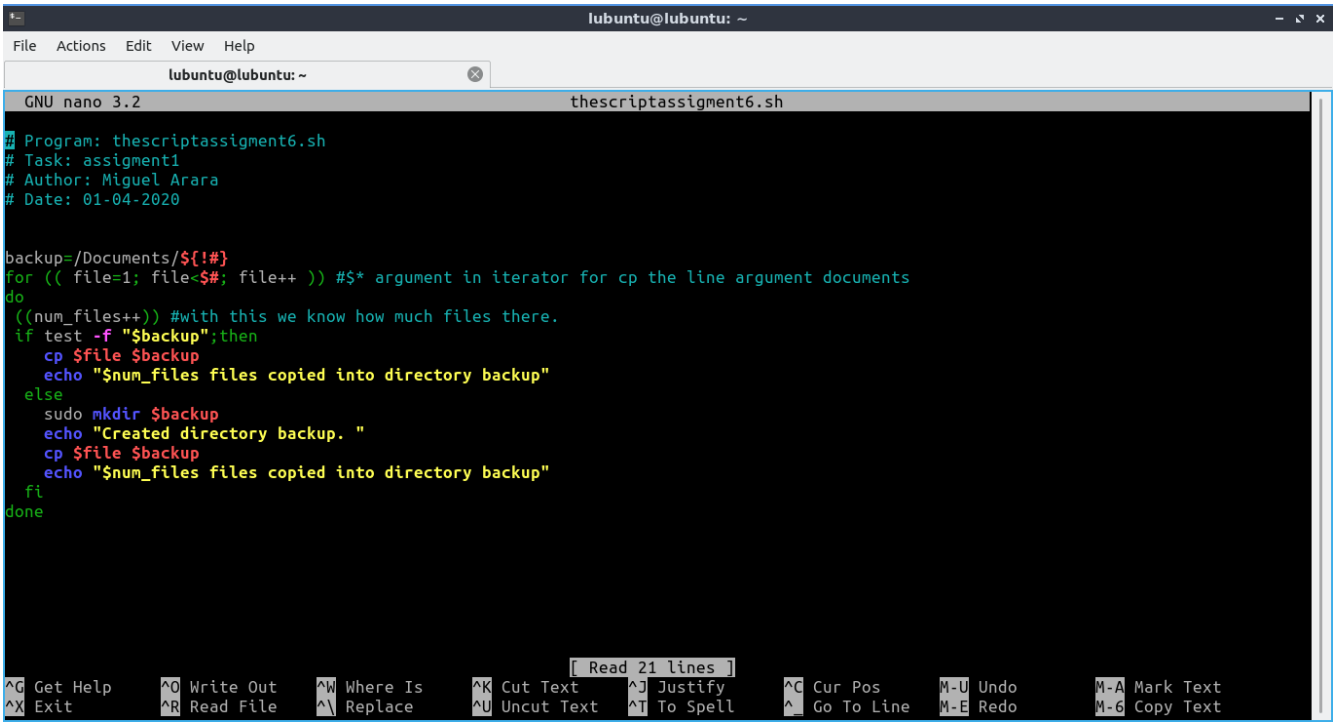
```
$ ls assignment6.sh dat1 dat2 dat3 dat4 dat5 dat6$
./assignment6.sh*backup

Created directory backup.7 files copied into directory
backup.

$ ls assignment6.sh backup dat1 dat2 dat3 dat4 dat5 dat6

$ ls backupassignment6.sh dat1 dat2 dat3 dat4 dat5 dat6
```

Upload instruction:Name the scriptassignment6.s



output: (I think that I have a little problem with privileges):