

1.Research how to create a hidden file and how to display it using ls command.

First of all, it is better to see where we are with `pwd` command.

```
yes@yes: ~$ pwd
/home/yes
yes@yes:~$
```

Secondly, we create a folder with make directory `mkdir` . We are going inside the directory with change directory `cd` Then, we can see what is going on inside the directory whether something is hidden or not with list all `ls -a`

```
yes@yes: ~/exercises$ ls -a
.  ..
yes@yes:~/exercises$
```

Now it is time to create hidden file with `touch .this-is-hidden-text` . Do you see there is a dot before the file that we want to create? When we type list command `ls` We cannot see the file we create because it is hidden. We should use list all `ls -a`

```
yes@yes: ~/exercises$ ls -a
.  ..
yes@yes:~/exercises$ touch .this-is-hidden.txt
yes@yes:~/exercises$ ls
yes@yes:~/exercises$
```

As you see below, after `ls -a` command we can see our hidden file. It cannot hide anymore.

```
yes@yes:~/exercises$ ls -a
.  ..
yes@yes:~/exercises$ touch .this-is-hidden.txt
yes@yes:~/exercises$ ls
yes@yes:~/exercises$ ls -a
.  .. .this-is-hidden.txt
yes@yes:~/exercises$
```

2. Research how to create multiple nested directories with one mkdir command. E.g. How would you create 'fun' directory along with all directories in the path below: /c/Users/unmesh/these/folders/are/just/for/fun

I didn't use `fun` directory in the example. I don't know it is going to be any problem or not. In order to create nested directories `mkdir -p` is used in the terminal. I will not explain how you can remove nested directories with one command because we haven't learned yet ☹

```
yes@yes: ~/exercises/folder1/folder2/folder3$ ls
yes@yes:~/exercises$ mkdir -p folder1/folder2/folder3
yes@yes:~/exercises$ ls
folder1
yes@yes:~/exercises$ cd folder1/folder2/folder3/
yes@yes:~/exercises/folder1/folder2/folder3$
```

3. Execute following commands terminal: echo "test" > fun echo "another test" >> fun wc -c fun Verify number of characters in the file.

We should go back to `exercises` folder. `cd ../../..` . Then i am going to test echo command.

```
yes@yes:~/exercises/folder1/folder2/folder3$ cd ../../..
yes@yes:~/exercises$
```

We are now in the `exercises` folder. I can now test the echo command `echo "test" > fun` and `echo "another test" >> fun` Then we count the words inside the file with `wc -c fun` . Actually, when we count the letters of those words it is not 18. I think spaces after the words are counted, but i'm not sure. ☹

```
yes@yes:~/exercises$ echo "test" > fun
yes@yes:~/exercises$ wc -c fun
5 fun
yes@yes:~/exercises$ echo "another test" >> fun
yes@yes:~/exercises$ wc -c fun
18 fun
yes@yes:~/exercises$ cat fun
test
another test
yes@yes:~/exercises$
```

4. Research how to append something in the file \*WITHOUT\* a newline character. E.g. echo "test" > fun echo "another test" >> fun cat fun Output of "cat fun" is test another test How would you use echo command, so that output would be: testanother test

I got tired. 🌀 Please just look at the image below. You will see how we can add words without a new line character. `echo -n` .

```
yes@yes:~/exercises$ echo -n "test" >> fun2
yes@yes:~/exercises$ echo -n " another test" >> fun2
yes@yes:~/exercises$ cat fun2
yes@yes:~/exercises$ cat fun2
test another testyes@yes:~/exercises$
```