

My Name :Sara Gamal Mohammed

1. Create a script that asks for user name then send a greeting to him.

```
1 echo please enter your name
2 read name
3 if [[ $name != "" ]]
4 then
5 echo Hello sir : $name , How are you
6 fi
```

```
sara@sara-Vostro-15-3510:~/homedir$ ./awkfile.sh
please enter your name
Sara
Hello sir : Sara , How are you
sara@sara-Vostro-15-3510:~/homedir$
```

2. Create a script called s1 that calls another script s2 where:
- In s1 there is a variable called x, it's value 5
 - Try to print the value of x in s2 by two different ways.

First way

```
s1 ~/homedir
1 export x=5
2 ./s2
```

```
s2 ~/homedir
1 echo $x
```

```
sara@sara-Vostro-15-3510:~/homedir$ ./s1
5
sara@sara-Vostro-15-3510:~/homedir$
```

Second way

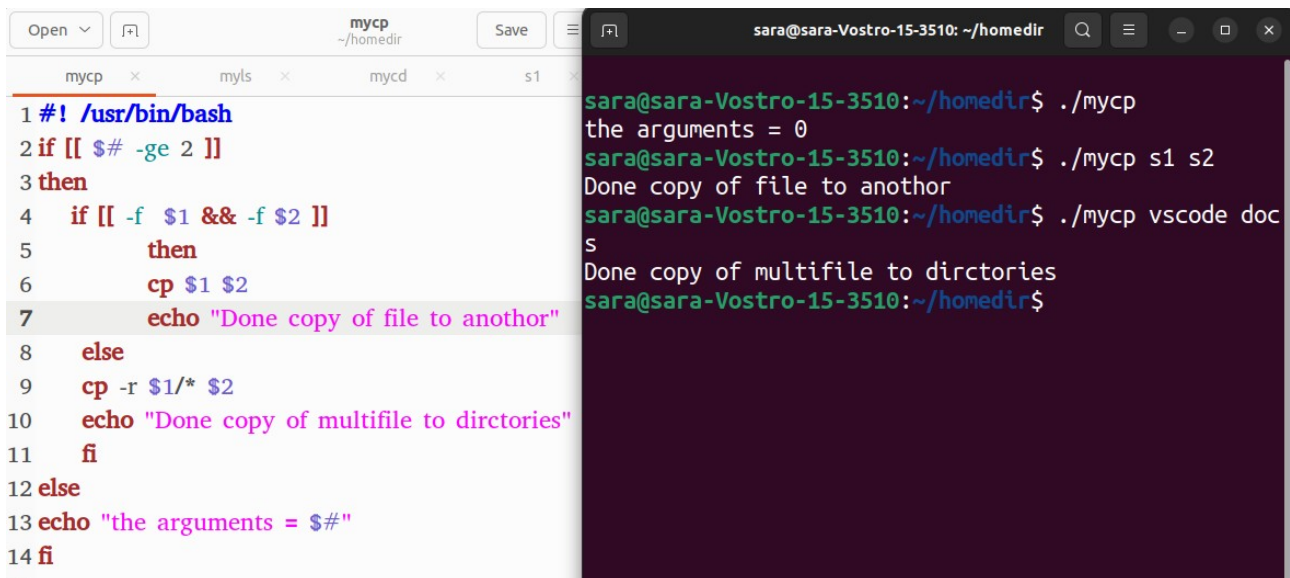
```
s1 ~/homedir
1 x=5
2 ./s2
```

```
s2 ~/homedir
1 echo $x
```

```
sara@sara-Vostro-15-3510:~/homedir$ ./s1
5
sara@sara-Vostro-15-3510:~/homedir$
```

3. Create a script called mycp where:

- It copies a file to another
- It copies multiple files to a directory.



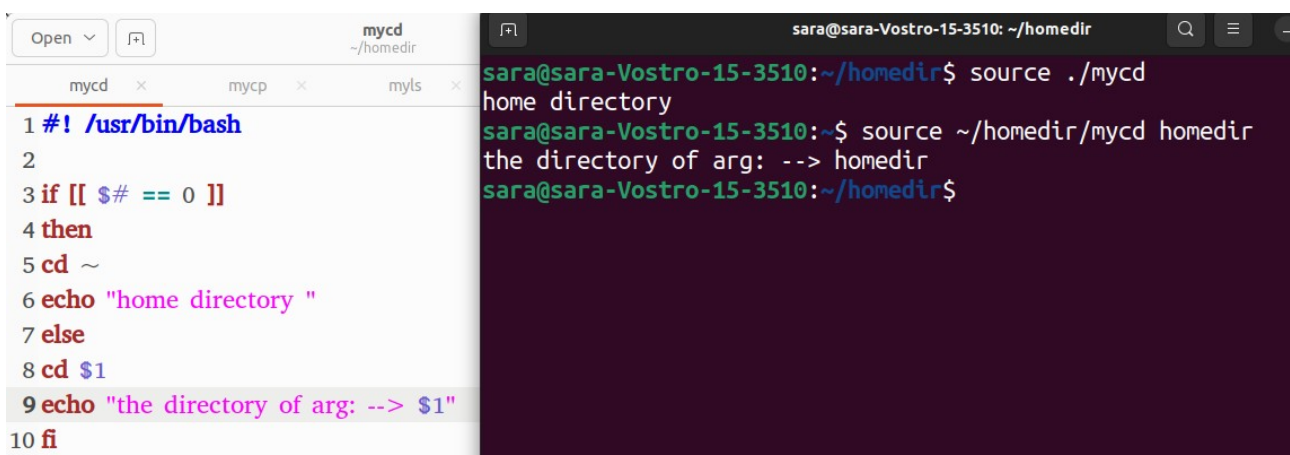
The screenshot shows a code editor on the left with the script `mycp` and a terminal on the right showing its execution. The script `mycp` is a bash script that checks the number of arguments. If there are 2 arguments, it copies a file. If there are more than 2 arguments, it copies multiple files to a directory. The terminal shows the script being run with no arguments, with 2 arguments (`s1 s2`), and with multiple arguments (`vscode docs`).

```
1 #!/usr/bin/bash
2 if [[ $# -ge 2 ]]
3 then
4     if [[ -f $1 && -f $2 ]]
5     then
6         cp $1 $2
7         echo "Done copy of file to another"
8     else
9         cp -r $1/* $2
10        echo "Done copy of multifile to directories"
11    fi
12 else
13     echo "the arguments = $#"
```

```
sara@sara-Vostro-15-3510: ~/homedir$ ./mycp
the arguments = 0
sara@sara-Vostro-15-3510:~/homedir$ ./mycp s1 s2
Done copy of file to another
sara@sara-Vostro-15-3510:~/homedir$ ./mycp vscode docs
Done copy of multifile to directories
sara@sara-Vostro-15-3510:~/homedir$
```

4. Create a script called mycd where:

- It changed directory to the user home directory, if it is called without arguments.
- Otherwise, it change directory to the given directory.



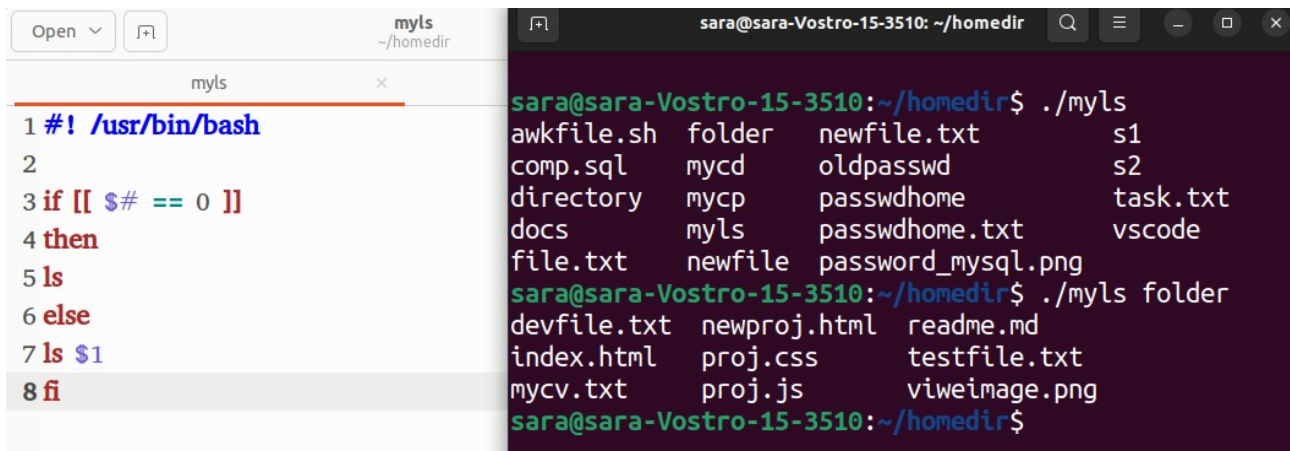
The screenshot shows a code editor on the left with the script `mycd` and a terminal on the right showing its execution. The script `mycd` is a bash script that checks if the number of arguments is 0. If so, it changes the directory to the user's home directory. Otherwise, it changes the directory to the given directory. The terminal shows the script being run with no arguments and with an argument (`homedir`).

```
1 #!/usr/bin/bash
2
3 if [[ $# == 0 ]]
4 then
5     cd ~
6     echo "home directory"
7 else
8     cd $1
9     echo "the directory of arg: --> $1"
10 fi
```

```
sara@sara-Vostro-15-3510:~/homedir$ source ./mycd
home directory
sara@sara-Vostro-15-3510:~$ source ~/homedir/mycd homedir
the directory of arg: --> homedir
sara@sara-Vostro-15-3510:~/homedir$
```

5. Create a script called myls where:

- It lists the current directory, if it is called without arguments.
- Otherwise, it lists the given directory.



The image shows a code editor on the left and a terminal window on the right. The code editor displays the content of a file named `mysls` located at `~/homedir`. The script is a Bash script that checks if it was called with arguments. If no arguments are provided, it runs `ls` in the current directory. If arguments are provided, it runs `ls` on the first argument.

```
1 #!/usr/bin/bash
2
3 if [[ $# == 0 ]]
4 then
5 ls
6 else
7 ls $1
8 fi
```

The terminal window shows the execution of the script. The prompt is `sara@sara-Vostro-15-3510: ~/homedir`. The user runs `./mysls`, and the terminal displays a list of files and directories in the current directory. The user then runs `./mysls folder`, and the terminal displays a list of files and directories in the `folder` directory.

```
sara@sara-Vostro-15-3510: ~/homedir$ ./mysls
awkfile.sh  folder  newfile.txt  s1
comp.sql    mycd    oldpasswd    s2
directory   mycp    passwdhome   task.txt
docs        myls    passwdhome.txt  vscode
file.txt    newfile  password_mysql.png
sara@sara-Vostro-15-3510: ~/homedir$ ./mysls folder
devfile.txt  newproj.html  readme.md
index.html   proj.css      testfile.txt
mycv.txt     proj.js       viweimage.png
sara@sara-Vostro-15-3510: ~/homedir$
```

6. Enhance the above script to support the following options individually:

- a. `-l`: list in long format
- b. `-a`: list all entries including the hiding files.
- c. `-d`: if an argument is a directory, list only its name
- d. `-i`: print inode number
- e. `-R`: recursively list subdirectories

```
mycd x mycp x myls x
1 #!/usr/bin/bash
2
3 if [[ $# != 0 ]]
4 then
5 echo "-----the output of option -l-----"
6 ls -l $1
7 echo "-----the output of option -a-----"
8 ls -a $1
9 echo "-----the output of option -d-----"
10 ls -d $1
11 echo "-----the output of option -i-----"
12 ls -i $1
13 echo "-----the output of option -R-----"
14 ls -R $1
15 else
16 echo "-----the output of option -l-----"
17 ls -l
18 echo "-----the output of option -a-----"
19 ls -a
20 echo "-----the output of option -d-----"
21 ls -d
22 echo "-----the output of option -i-----"
23 ls -i
24 echo "-----the output of option -R-----"
25 ls -R
26 fi
```

```
sara@sara-Vostro-15-3510:~/homedir$ ./mysl vscode
-----the output of option -l-----
total 1644
-rw-rw-r-- 1 sara sara      0 14:31 28  يـون devfile.txt
-rw-rw-r-- 1 sara sara      0 22:35 2  يـون index.html
-rw-rw-r-- 1 sara sara    366 14:31 28  يـون newproj.html
-rw-rw-r-- 1 sara sara     31 13:49 28  يـون proj.css
-rw-rw-r-- 1 sara sara     45 13:49 28  يـون proj.js
-rw-rw-r-- 1 sara sara   1084 16:21 28  يـون readme.md
-rw-rw-r-- 1 sara sara      0 14:38 28  يـون testfile.txt
-rw-rw-r-- 1 sara sara 1665571 16:21 28  يـون viweimage.png
-----the output of option -a -----
.  devfile.txt index.html proj.css readme.md viweimage.png
.. .git          newproj.html proj.js testfile.txt
-----the output of option -d -----
vscode
-----the output of option -i -----
5650829 devfile.txt 5649293 newproj.html 5639986 proj.js
5669264 index.html 5642387 proj.css      5649603 readme.md
-----the output of option -R -----
vscode:
devfile.txt index.html newproj.html proj.css proj.js read
sara@sara-Vostro-15-3510:~/homedir$
```