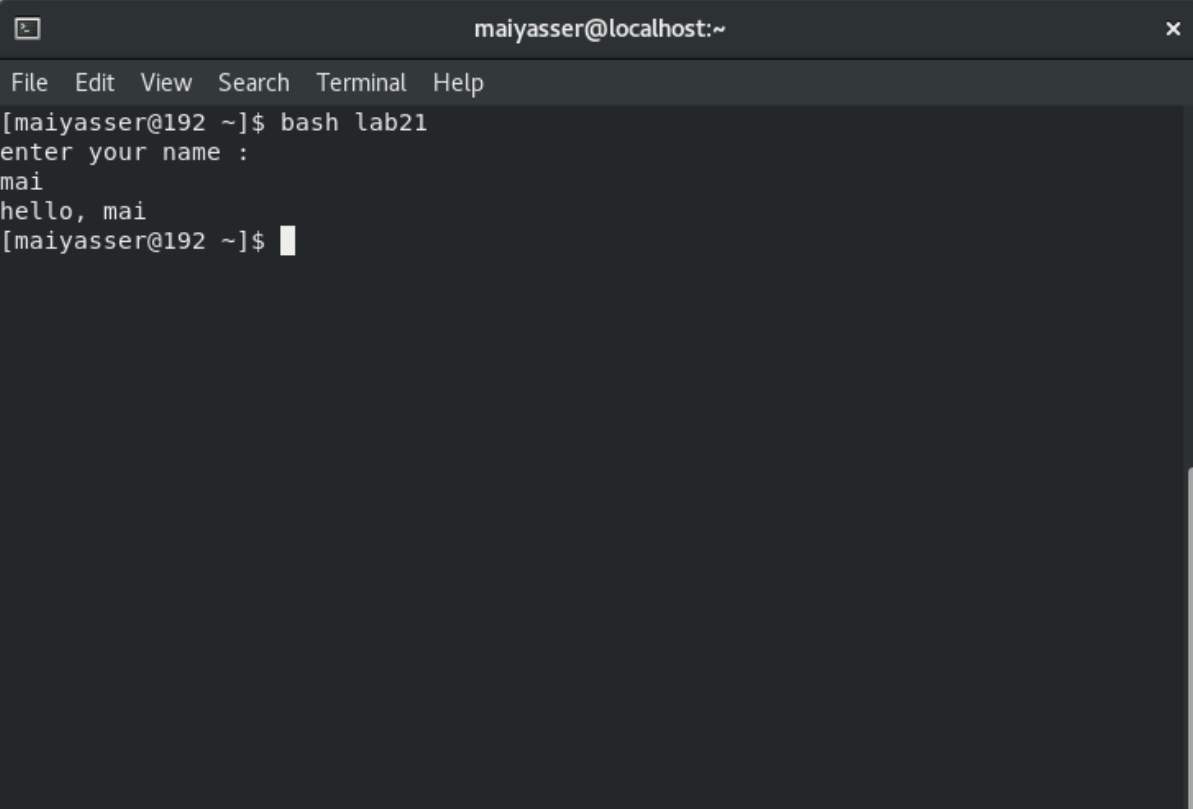


1. Create a script that asks for user name then send a greeting to him.



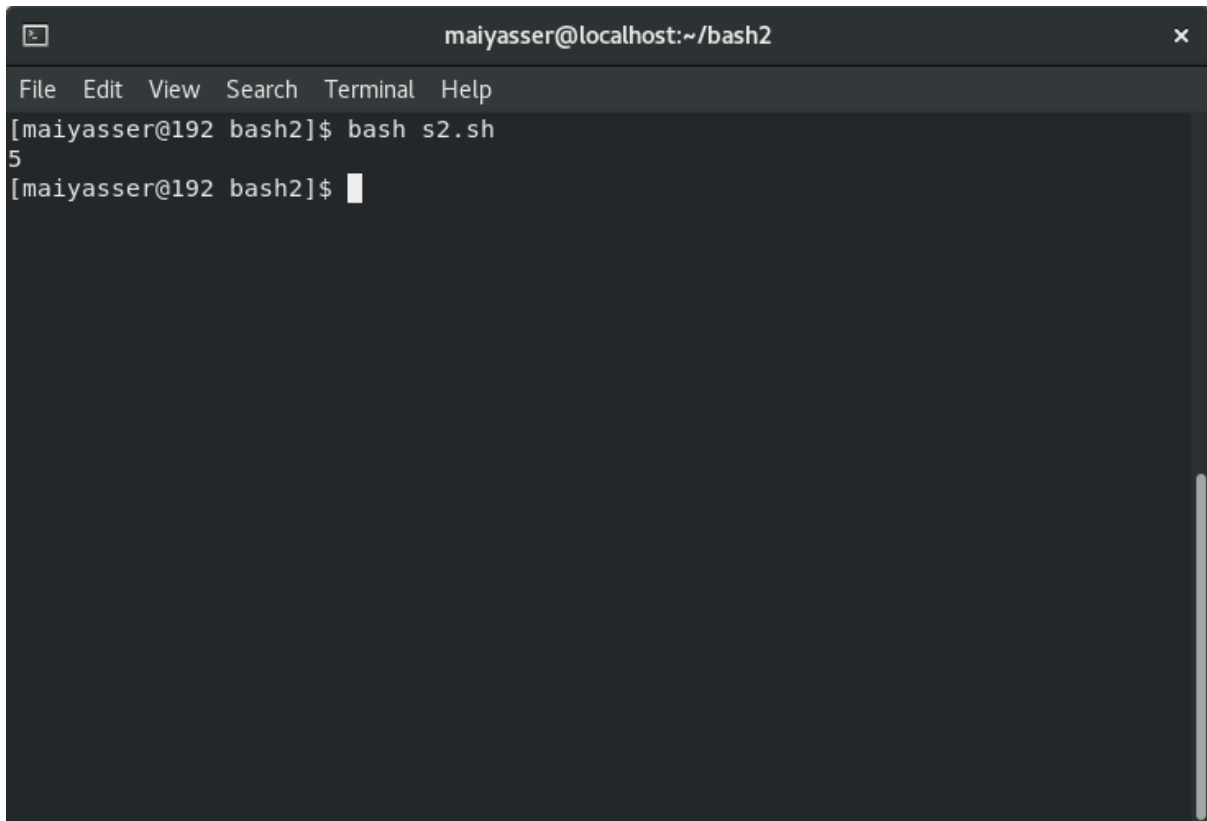
The screenshot shows a terminal window titled "maiyasser@localhost:~". The window has a menu bar with "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a user running a script named "lab21". The script prompts the user to "enter your name :", the user enters "mai", and the script outputs "hello, mai". The terminal ends with the prompt "[maiyasser@192 ~]\$".

```
maiyasser@localhost:~  
File Edit View Search Terminal Help  
[maiyasser@192 ~]$ bash lab21  
enter your name :  
mai  
hello, mai  
[maiyasser@192 ~]$
```

2. Create a script called s1lab 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.

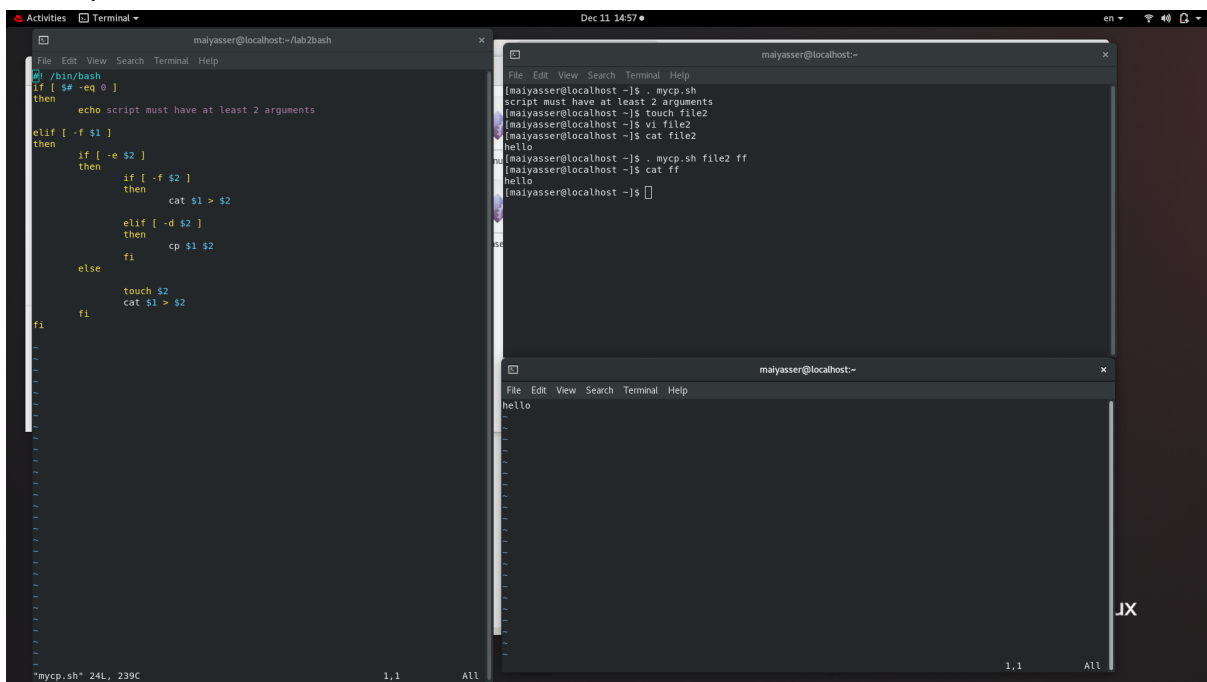
[illegible]



```
maiyasser@localhost:~/bash2
File Edit View Search Terminal Help
[maiyasser@192 bash2]$ bash s2.sh
5
[maiyasser@192 bash2]$
```

3. Create a script called mycp where:

a. It copies a file to another



```
maiyasser@localhost:~/lab2bash
File Edit View Search Terminal Help
if [ $# -eq 0 ]
then
    echo script must have at least 2 arguments
elif [ -f $1 ]
then
    if [ -e $2 ]
    then
        if [ -f $2 ]
        then
            cat $1 > $2
        elif [ -d $2 ]
        then
            cp $1 $2
        fi
    else
        touch $2
        cat $1 > $2
    fi
fi

maiyasser@localhost ~$ . mycp.sh
script must have at least 2 arguments
maiyasser@localhost ~$ touch file2
maiyasser@localhost ~$ vi file2
maiyasser@localhost ~$ cat file2
hello
maiyasser@localhost ~$ . mycp.sh file2 ff
maiyasser@localhost ~$ cat ff
hello
maiyasser@localhost ~$

maiyasser@localhost ~$
File Edit View Search Terminal Help
hello
```

b. It copies multiple files to a directory.

The screenshot shows a terminal window with two panes. The left pane shows the creation of a script named `mycp.sh` in the `~/lab2bash` directory. The script uses conditional logic to handle different argument counts and file types. The right pane shows the execution of the script with various arguments, demonstrating its functionality. The terminal output shows the script's behavior for different inputs, including file creation, directory handling, and file copying.

```
#!/bin/bash
if [ $# -eq 0 ]
then
    echo script must have at least 2 arguments
elif [ -f $1 ]
then
    if [ -e $2 ]
    then
        if [ -f $2 ]
        then
            cat $1 > $2
        elif [ -d $2 ]
        then
            cp $1 $2
        else
            touch $2
            cat $1 > $2
        fi
    fi
fi
if [ $# -gt 2 ]
then
    cp $*
```

```
[maiya@localhost ~]$ mkdir mycpTest
[maiya@localhost ~]$ cat file2
hello
[maiya@localhost ~]$ cat ff
hello
[maiya@localhost ~]$ . mycp.sh ff file2 mycpTest
[maiya@localhost ~]$ cat mycpTest
cat: mycpTest: is a directory
[maiya@localhost ~]$ ls mycpTest
ff file2
[maiya@localhost ~]$ cd mycpTest
[maiya@localhost mycpTest]$ cat ff
hello
[maiya@localhost mycpTest]$
```

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 terminal window with two panes. The left pane shows the execution of the `mycd.sh` script in the `/etc` directory. The right pane shows the creation of the `mycd.sh` script in the `~/lab2bash` directory. The script uses conditional logic to change the directory to the user home directory if no arguments are provided, or to the given directory if arguments are provided. The terminal output shows the script's behavior for different inputs, including directory changes.

```
#!/bin/bash
if [ $# -eq 0 ]
then
    cd
else
    cd $1
fi
```

```
[maiya@localhost ~]$ . mycd.sh lab2bash
[maiya@localhost lab2bash]$ . mycd.sh
[maiya@localhost ~]$ . mycd.sh /etc
[maiya@localhost etc]$
```

5. Create a script called myls where:
 - a. It lists the current directory, if it is called without arguments.
 - b. Otherwise, it lists the given directory.

```

File Edit View Search Terminal Help
/bin/bash
if [ $# -eq 0 ]
then
ls
else
if [ -d $1 ]
then
ls $1
else
echo argument must be a directory
fi
fi
fi

~mysl.sh* 12L, 120C      1,1      All

```

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

```

File Edit View Search Terminal Help
/bin/bash
if [ $# -eq 0 ]
then
ls
else
if [ $1 = '-l' ]
then
echo list in long format
elif [ $1 = '-a' ]
then
echo list all entries including the hiding files
elif [ $1 = '-d' ]
then
echo if an argument is a directory list only its name
elif [ $1 = '-i' ]
then
echo print inode number
elif [ $1 = '-R' ]
then
echo recursively list subdirectories
fi
fi
fi

~mysl.sh* 12L, 120C      1,1      All

```

7. Create a script called mytest where:

- It check the type of the given argument (file/directory)
- It check the permissions of the given argument (read/write/execute)

```
#!/bin/bash
if [ $# -gt 0 ]
then
    if [ -f $1 ]
    then
        echo this is a file
        if [ -r $1 ]
        then
            echo this file is readable
        elif [ -w $1 ]
        then
            echo this file is writable
        elif [ -x $1 ]
        then
            echo this file is executable
        fi
    elif [ -d $1 ]
    then
        echo this is a directory
        if [ -r $1 ]
        then
            echo this directory is readable
        elif [ -w $1 ]
        then
            echo this directory is writable
        elif [ -x $1 ]
        then
            echo this directory is executable
        fi
    fi
else
    echo you must entre an argument
fi
```

mytest.sh 33L, 520C

8. Create a script called myinfo where:

- It asks the user about his/her logname.
- It print full info about files and directories in his/her home directory
- Copy his/her files and directories as much as you can in /tmp directory.
- Gets his current processes status.

```
#!/bin/bash
echo "what is your login name"
read $ans
home=`grep ^$ans /etc/passwd | cut -d: -f6`
if [ $home ]
then
    ls -l $home
    cp $home/* /tmp
    ps -u $answer
else
    echo "username not exist"
fi
```

myinfo.sh 16L, 201C