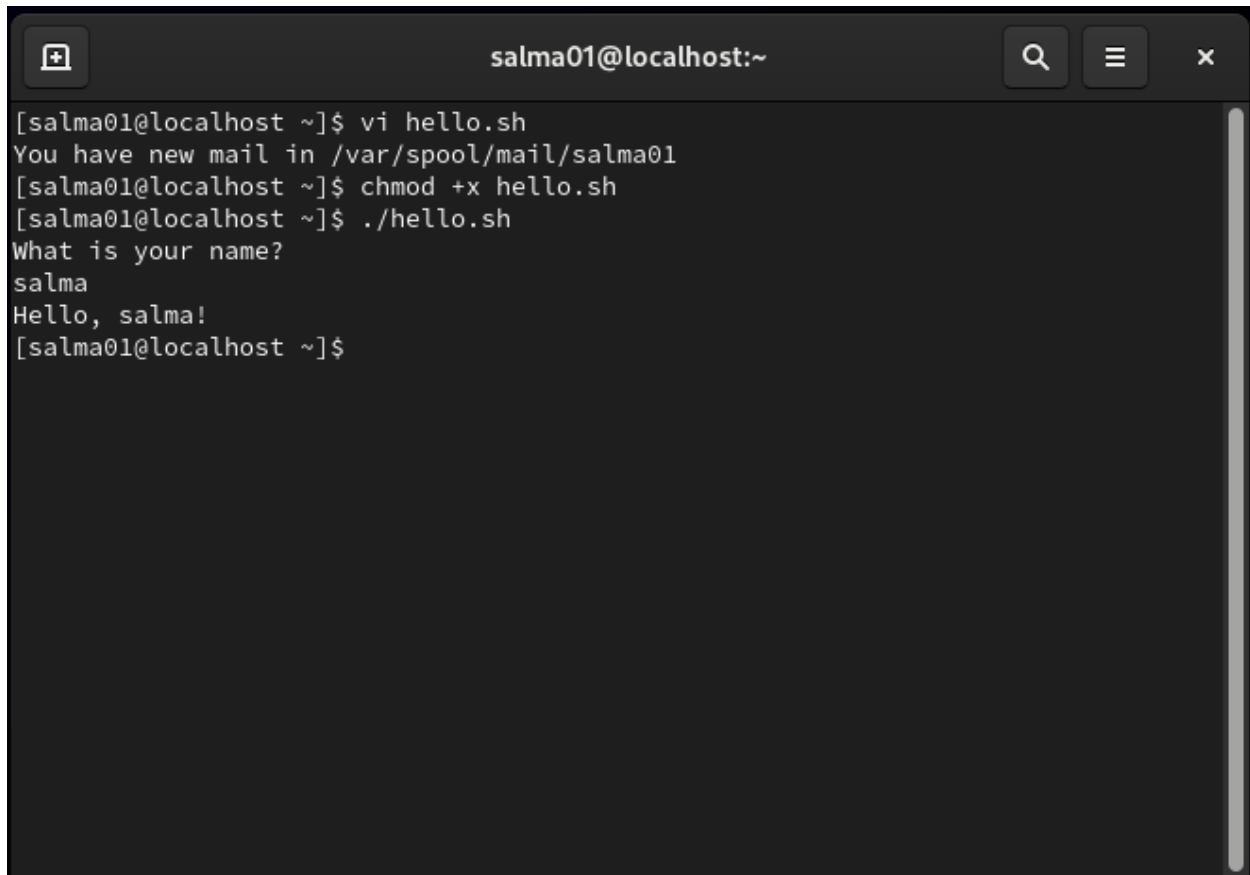


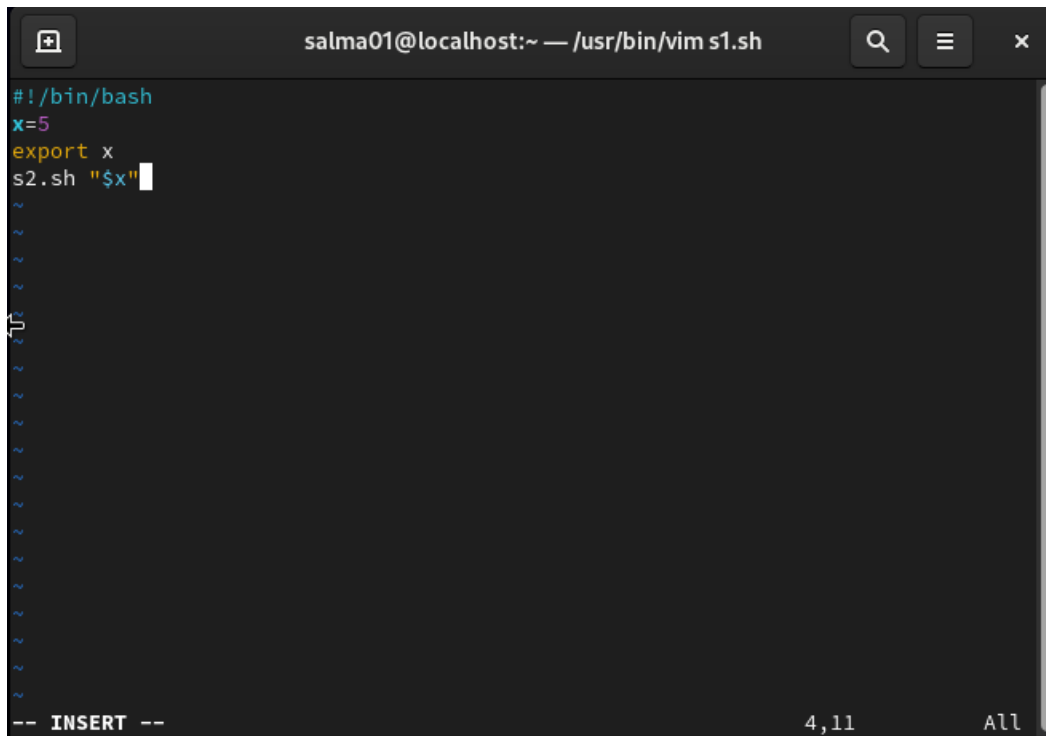
1. Create a script that asks for user name then send a greeting to him.

A terminal window titled 'salma01@localhost:~' with search, menu, and close icons in the title bar. The terminal shows the following sequence of commands and output:

```
[salma01@localhost ~]$ vi hello.sh
You have new mail in /var/spool/mail/salma01
[salma01@localhost ~]$ chmod +x hello.sh
[salma01@localhost ~]$ ./hello.sh
What is your name?
salma
Hello, salma!
[salma01@localhost ~]$
```

2. Create a script called s1 that calls another script s2 where:

a. In s1 there is a variable called x, it's value 5

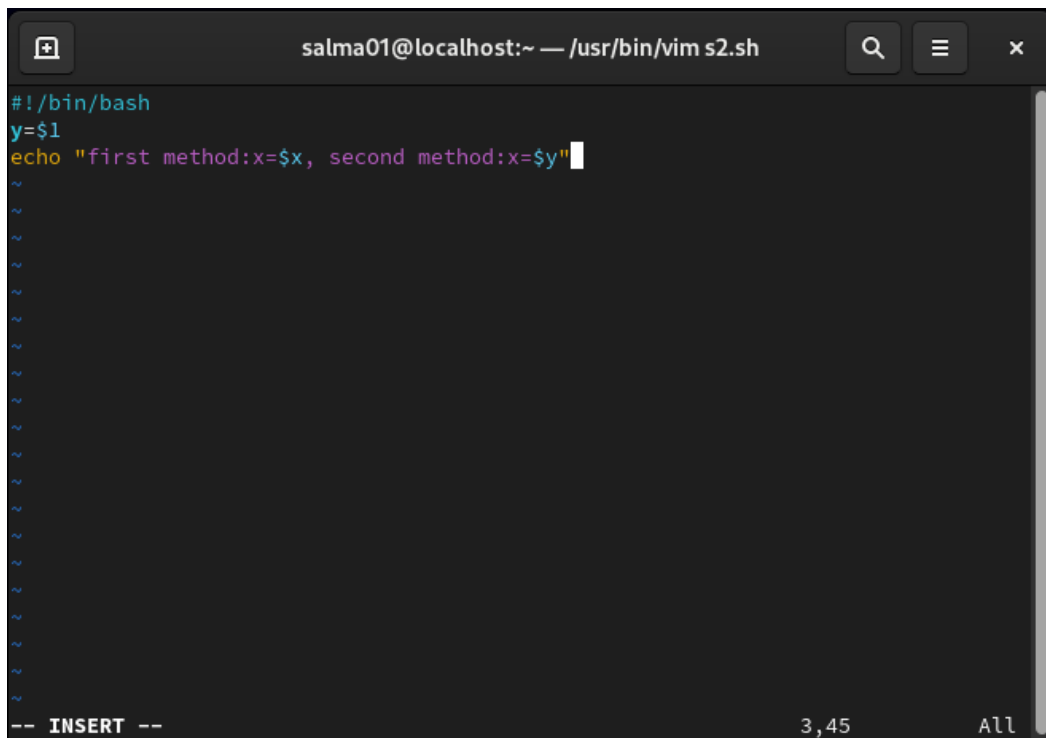


The screenshot shows a terminal window with the title bar "salma01@localhost:~ — /usr/bin/vim s1.sh". The editor displays the following code:

```
#!/bin/bash
x=5
export x
s2.sh "$x"
```

The status bar at the bottom indicates "-- INSERT --", the cursor position "4,11", and the search status "All".

b. Try to print the value of x in s2 by two different ways.




The screenshot shows a terminal window with the title bar "salma01@localhost:~ — /usr/bin/vim s2.sh". The editor displays the following code:

```
#!/bin/bash
y=$1
echo "first method:x=$x, second method:x=$y"
```

The status bar at the bottom indicates "-- INSERT --", the cursor position "3,45", and the search status "All".

b. It copies multiple files to a directory.



A terminal window titled "salma01@localhost:~" with search, menu, and close buttons. The terminal shows the execution of a shell script named "mycp.sh" with arguments "s1.sh" and "s2.sh". The prompt changes from "~" to "~]" during execution and back to "~" when it completes.

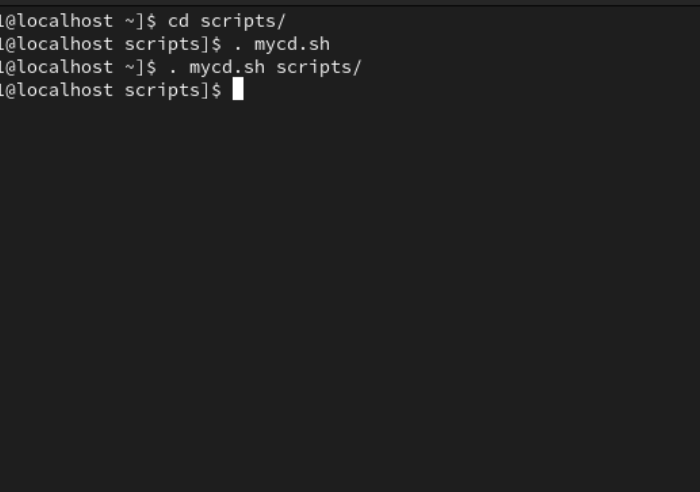
```
salma01@localhost:~$ mycp.sh s1.sh s2.sh
[salma01@localhost ~]$
```

```

salma01@localhost:~ — /usr/bin/vim mycp.sh
#!/bin/bash
cp "$1" "$2"
cp "$@" scripts

```

"mycp.sh" 5L, 43B 1,1 All

- 
- A terminal window with a dark background. The title bar at the top shows a file icon, the text 'salma01@localhost:~/scripts', and three buttons: a magnifying glass, a hamburger menu, and a close 'x' button. The terminal content shows a sequence of commands and their outputs:
- ```
[salma01@localhost ~]$ cd scripts/
[salma01@localhost scripts]$. mycd.sh
[salma01@localhost ~]$. mycd.sh scripts/
[salma01@localhost scripts]$
```
- A white cursor is visible at the end of the last line. A vertical scrollbar is on the right side of the terminal area.

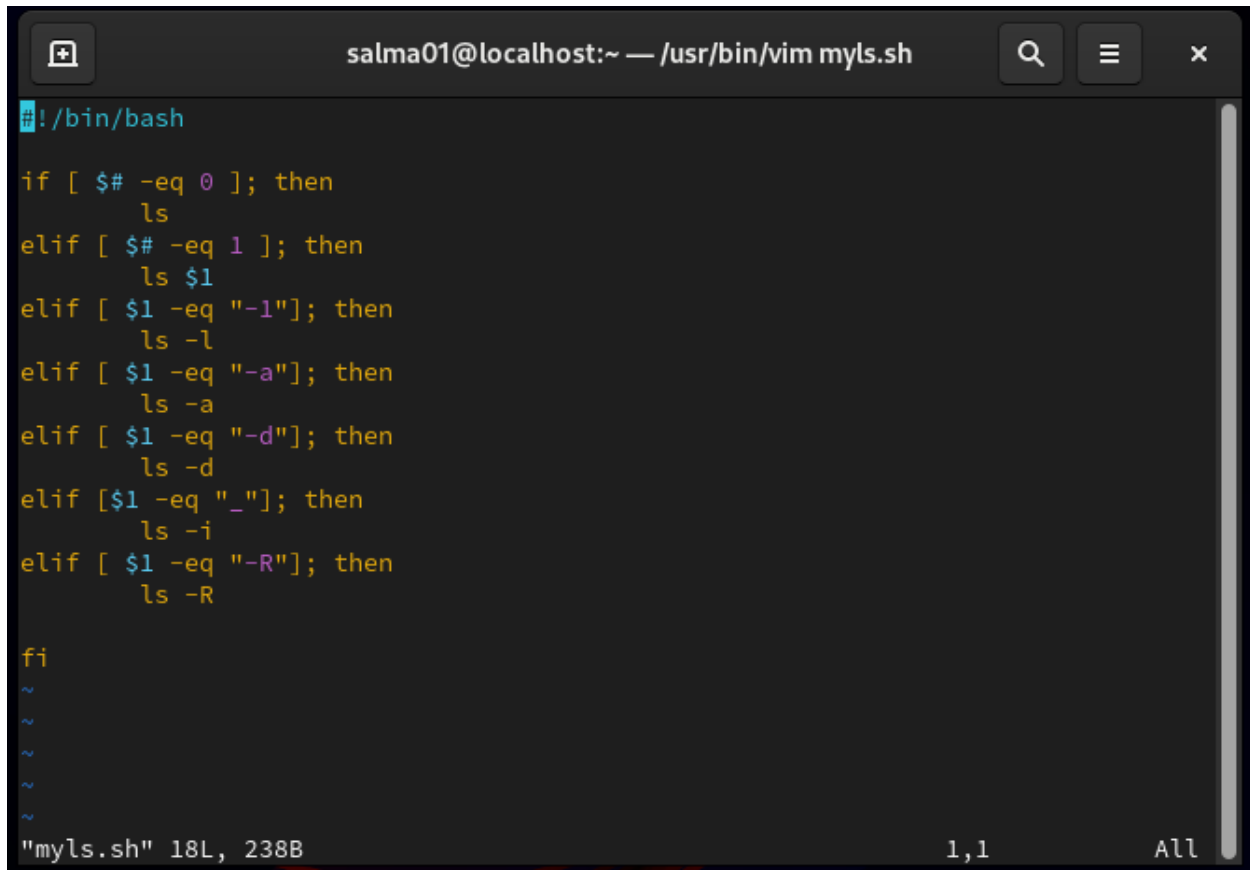
[illegible]

b. Otherwise, it lists the given directory.

[illegible]

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

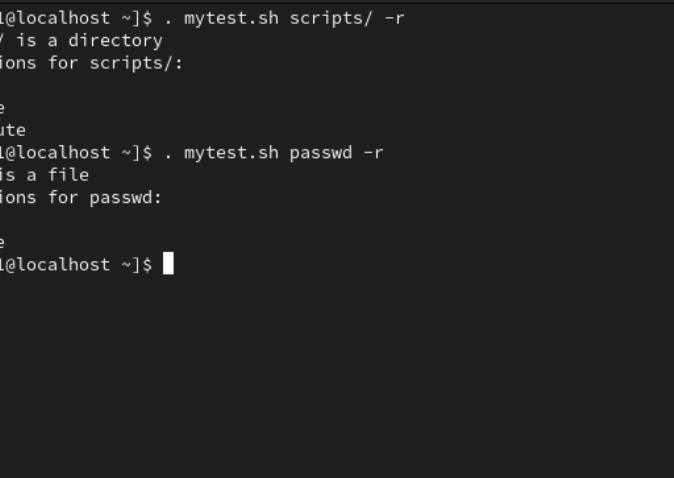


```
salma01@localhost:~ — /usr/bin/vim myls.sh
#!/bin/bash

if [$# -eq 0]; then
 ls
elif [$# -eq 1]; then
 ls $1
elif [$1 -eq "-l"]; then
 ls -l
elif [$1 -eq "-a"]; then
 ls -a
elif [$1 -eq "-d"]; then
 ls -d
elif [$1 -eq "-i"]; then
 ls -i
elif [$1 -eq "-R"]; then
 ls -R
fi

~
~
~
~
~
~
"mysls.sh" 18L, 238B 1,1 All
```

b. It check the permissions of the given argument (read/write/execute)



```
salma01@localhost:~
[salma01@localhost ~]$. mytest.sh scripts/ -r
scripts/ is a directory
Permissions for scripts/:
is Read
is Write
is Execute
[salma01@localhost ~]$. mytest.sh passwd -r
passwd is a file
Permissions for passwd:
is Read
is Write
[salma01@localhost ~]$
```

A screenshot of a terminal window titled "salma01@localhost:~ — /usr/bin/vim mytest.sh". The terminal shows the execution of a shell script named "mytest.sh". The script starts with a shebang "#!/bin/bash". It uses an if-elif-else-fi structure to check if a file exists, if it's a directory, and its permissions. The output of the script is displayed in green text. At the bottom of the terminal, the command prompt shows "mytest.sh" followed by "14L, 268B", indicating the current position in the file. On the right side of the terminal, there are search and navigation icons, and the text "1,1 All" is visible.

```
#!/bin/bash

if [-f "$1"]; then
 echo "$1 is a file"
elif [-d "$1"]; then
 echo "$1 is a directory"
else
 echo "Not a file or a directory"
fi

echo "Permissions for $1:"
[-r "$1"] && echo "is Read"
[-w "$1"] && echo "is Write"
[-x "$1"] && echo "is Execute"

~
~
~
~
~
~
~
~
~
~
```

"mytest.sh" 14L, 268B 1,1 All

d. Gets his current processes status.

[illegible]