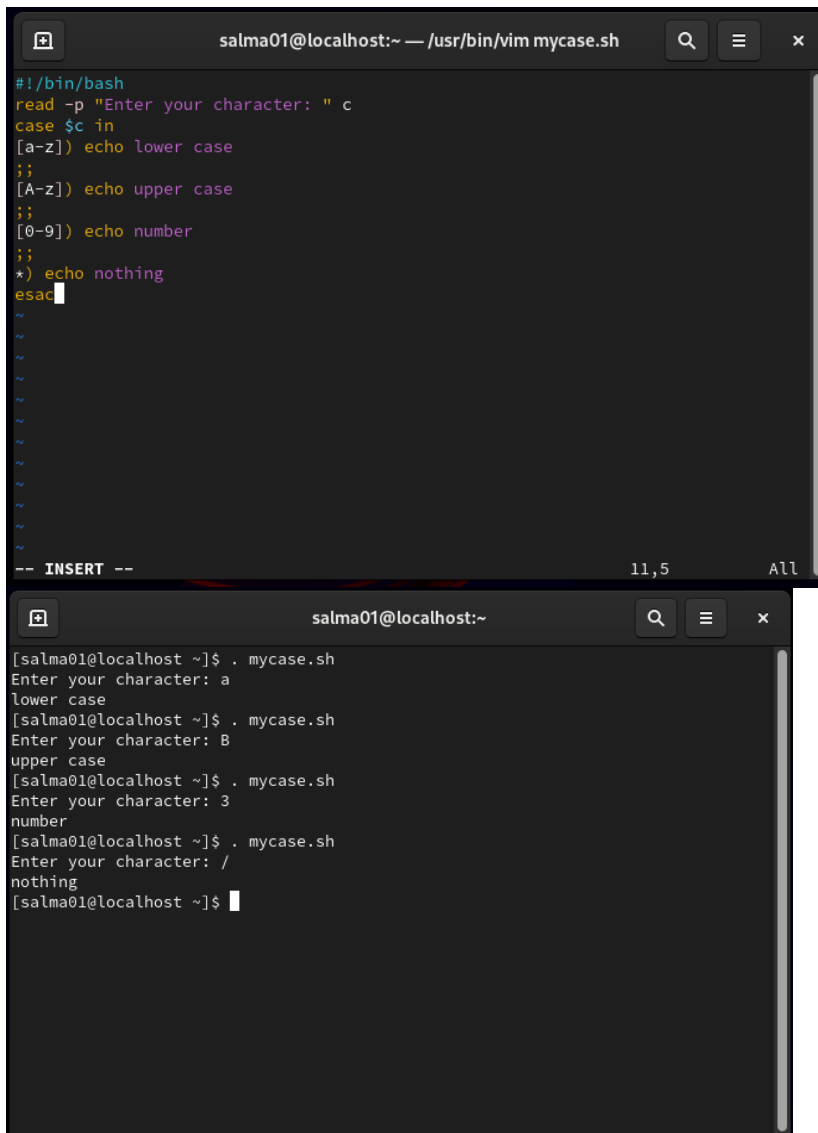




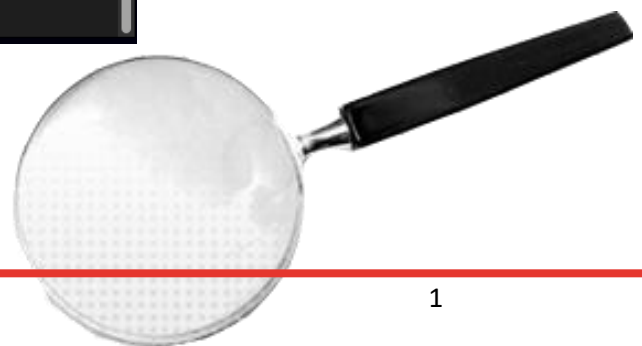
1. Write a script called mycase, using the case utility to checks the type of character entered by a user:
 - a. Upper Case.
 - b. Lower Case.
 - c. Number.
 - d. Nothing.



```
salma01@localhost:~ — /usr/bin/vim mycase.sh
#!/bin/bash
read -p "Enter your character: " c
case $c in
[a-z]) echo lower case
;;
[A-Z]) echo upper case
;;
[0-9]) echo number
;;
*) echo nothing
esac

-- INSERT -- 11,5 All

salma01@localhost:~
[salma01@localhost ~]$ . mycase.sh
Enter your character: a
lower case
[salma01@localhost ~]$ . mycase.sh
Enter your character: B
upper case
[salma01@localhost ~]$ . mycase.sh
Enter your character: 3
number
[salma01@localhost ~]$ . mycase.sh
Enter your character: /
nothing
[salma01@localhost ~]$
```



2. Enhanced the previous script, by checking the type of string entered by a user.
 - a. Upper Cases.
 - b. Lower Cases.
 - c. Numbers.
 - d. Mix.
 - e. Nothing.

```
salma01@localhost:~ — /usr/bin/vim mycase.sh
```

```
#!/bin/bash
read -p "Enter your character: " c
case $c in
+([a-z]) ) echo lower case
;;
+([A-Z]) ) echo upper case
;;
+([0-9]) ) echo number
;;
+([a-zA-Z]) ) echo mix
;;
+(*) ) echo nothing
esac

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

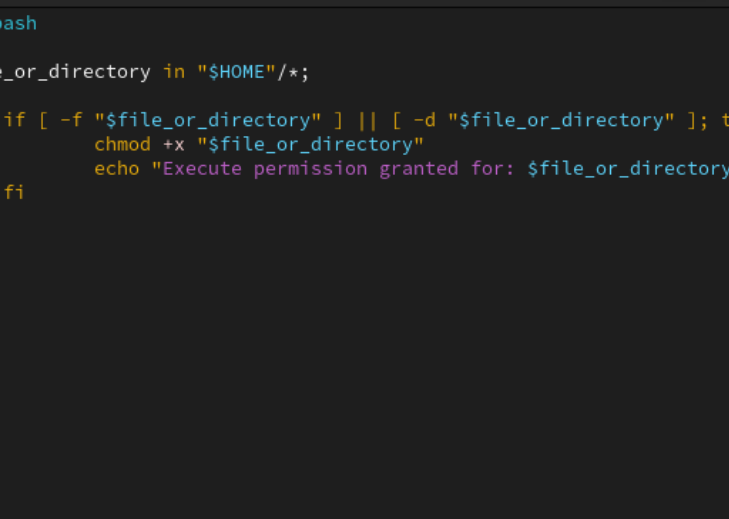
```
"mycase.sh" 13L, 197B
```

```
1,1 All
```

```
salma01@localhost:~  
[salma01@localhost ~]$ . mycase.sh  
Enter your character: SALMA  
upper case  
[salma01@localhost ~]$ . mycase.sh  
Enter your character: salma  
lower case  
[salma01@localhost ~]$ . mycase.sh  
Enter your character: 1234  
number  
[salma01@localhost ~]$ . mycase.sh  
Enter your character: SALMASalma  
mix  
[salma01@localhost ~]$ . mycase.sh  
Enter your character: ////  
nothing  
[salma01@localhost ~]$
```



3. Write a script called mychmod using for utility to give execute permission to all files and directories in your home directory.

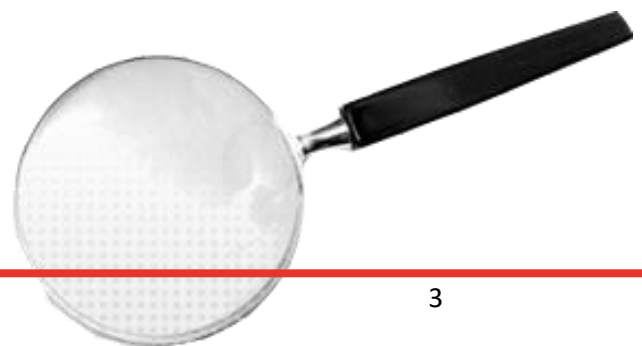


```
#!/bin/bash

for file_or_directory in "$HOME"/*;
do
    if [ -f "$file_or_directory" ] || [ -d "$file_or_directory" ]; then
        chmod +x "$file_or_directory"
        echo "Execute permission granted for: $file_or_directory"
    fi
done
```

1,1 All

```
salma01@localhost:~$ . mychmod.sh
Execute permission granted for: /home/salma01/1
Execute permission granted for: /home/salma01/commands
Execute permission granted for: /home/salma01/Desktop
Execute permission granted for: /home/salma01/docs
Execute permission granted for: /home/salma01/Documents
Execute permission granted for: /home/salma01/Downloads
Execute permission granted for: /home/salma01/email
Execute permission granted for: /home/salma01/error
Execute permission granted for: /home/salma01/f1
Execute permission granted for: /home/salma01/f2
Execute permission granted for: /home/salma01/f8
Execute permission granted for: /home/salma01/file1
Execute permission granted for: /home/salma01/file2
Execute permission granted for: /home/salma01/hello.sh
Execute permission granted for: /home/salma01/ls_output
Execute permission granted for: /home/salma01/Music
Execute permission granted for: /home/salma01/mycase.sh
Execute permission granted for: /home/salma01/mycd.sh
Execute permission granted for: /home/salma01/mychmod.sh
Execute permission granted for: /home/salma01/mycp.sh
Execute permission granted for: /home/salma01/mycv
Execute permission granted for: /home/salma01/myinfo.sh
Execute permission granted for: /home/salma01/myls.sh
```



4. Write a script called mybackup using for utility to create a backup of only files in your home directory.

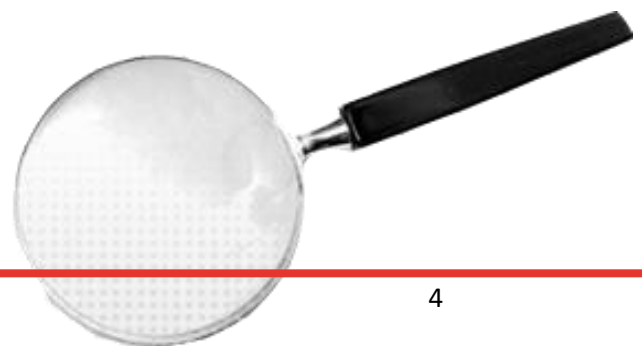
```
salma01@localhost:~ — /usr/bin/vim mybackup.sh
#!/bin/bash

backup_dir="$HOME/backup"
mkdir "$backup_dir" && echo "Backup directory created: $backup_dir"

for file in "$HOME"/*;
do
    if [ -f "$file" ]; then
        cp "$file" "$backup_dir"
        echo "File backed up: $file"
    fi
done

echo "Backup Done"
~
~
~
~
~
~
~
~
"mybackup.sh" 14L, 245B 1,1 All
```

```
salma01@localhost:~
[salma01@localhost ~]$ . mybackup.sh
mkdir: cannot create directory '/home/salma01/backup': File exists
File backed up: /home/salma01/1
File backed up: /home/salma01/backup_dir
File backed up: /home/salma01/commands
File backed up: /home/salma01/email
File backed up: /home/salma01/error
File backed up: /home/salma01/f1
File backed up: /home/salma01/f2
cp: cannot open '/home/salma01/file1' for reading: Permission denied
File backed up: /home/salma01/file1
File backed up: /home/salma01/file2
File backed up: /home/salma01/hello.sh
File backed up: /home/salma01/ls_output
File backed up: /home/salma01/mybackup.sh
File backed up: /home/salma01/mycase.sh
File backed up: /home/salma01/mycd.sh
File backed up: /home/salma01/mychmod.sh
File backed up: /home/salma01/mycp.sh
File backed up: /home/salma01/mycv
File backed up: /home/salma01/myinfo.sh
File backed up: /home/salma01/myls.sh
File backed up: /home/salma01/mytest.sh
File backed up: /home/salma01/oldpasswd
```



5. Write a script called mymail using for utility to send a mail to all users in the system.
Note: write the mail body in a file called mtemplate.

```
salma01@localhost:~ — /usr/bin/vim mymail.sh
#!/bin/bash
for user in $(getent passwd | cut -d: -f1)
do
    mail -s "email" "$user" < mtemplate
    echo "mail to $user was sent"
done
```

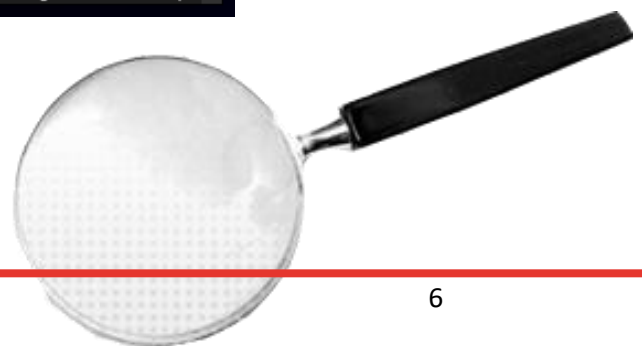
```
salma01@localhost:~
[salma01@localhost ~]$ . mymail.sh
mail to root was sent
mail to bin was sent
mail to daemon was sent
mail to adm was sent
mail to sync was sent
mail to shutdown was sent
mail to halt was sent
mail to mail was sent
mail to operator was sent
mail to games was sent
mail to ftp was sent
mail to nobody was sent
mail to systemd-coredump was sent
mail to dbus was sent
mail to polkitd was sent
mail to avahi was sent
mail to tss was sent
mail to colord was sent
mail to clevis was sent
mail to rtkit was sent
mail to sssd was sent
mail to geoclue was sent
mail to libstoragemgmt was sent
```



6. Write a script called chkmail to check for new mails every 10 seconds. Note: mails are saved in /var/mail/username.

```
salma01@localhost:~ — /usr/bin/vim chkmail.sh
#!/bin/bash
while true;
do
    if [ -s "/var/mai/salma01" ]; then
        echo "You have new mail!"
        mail
    fi
    sleep 10
done
```

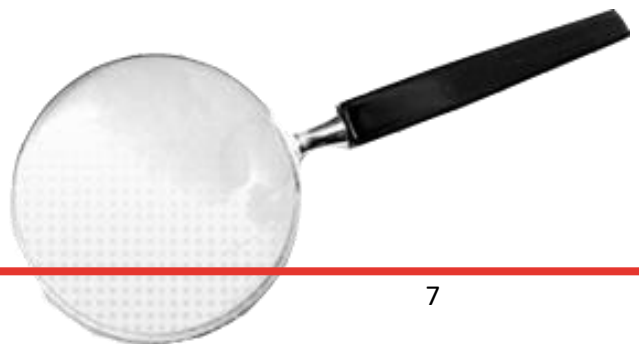
```
salma01@localhost:~
[salma01@localhost ~]$ . chkmail.sh
You have new mail!
s-nail version v14.9.22. Type '?' for help
/var/spool/mail/salma01: 156 messages 151 new 154 unread
 1 (Cron Daemon) 2023-11-21 17:32 33/1322 "Cron <salma01@localhost>"
U 2 (Cron Daemon) 2023-11-21 17:33 33/1321 "Cron <salma01@localhost>"
U 3 (Cron Daemon) 2023-11-21 17:34 35/1353 "Cron <salma01@localhost>"
 4 (Cron Daemon) 2023-11-21 17:35 33/1321 "Cron <salma01@localhost>"
U 5 (Cron Daemon) 2023-11-21 18:00 34/1314 "Cron <root@localhost> /"
•N 6 (Cron Daemon) 2023-11-22 16:30 33/1303 "Cron <root@localhost> /"
N 7 (Cron Daemon) 2023-11-22 16:40 33/1303 "Cron <root@localhost> /"
N 8 (Cron Daemon) 2023-11-22 16:50 33/1303 "Cron <root@localhost> /"
N 9 (Cron Daemon) 2023-11-22 17:00 33/1304 "Cron <root@localhost> /"
N 10 (Cron Daemon) 2023-11-22 17:10 33/1304 "Cron <root@localhost> /"
N 11 (Cron Daemon) 2023-11-22 17:20 33/1304 "Cron <root@localhost> /"
N 12 (Cron Daemon) 2023-11-22 17:30 33/1304 "Cron <root@localhost> /"
N 13 (Cron Daemon) 2023-11-22 17:40 33/1304 "Cron <root@localhost> /"
N 14 (Cron Daemon) 2023-11-22 17:50 33/1304 "Cron <root@localhost> /"
N 15 (Cron Daemon) 2023-11-22 18:00 33/1304 "Cron <root@localhost> /"
N 16 (Cron Daemon) 2023-11-22 18:10 33/1304 "Cron <root@localhost> /"
N 17 (Cron Daemon) 2023-11-22 18:20 33/1304 "Cron <root@localhost> /"
N 18 (Cron Daemon) 2023-11-22 18:30 33/1304 "Cron <root@localhost> /"
N 19 (Cron Daemon) 2023-11-22 18:40 33/1304 "Cron <root@localhost> /"
N 20 (Cron Daemon) 2023-11-22 18:50 33/1304 "Cron <root@localhost> /"
```



7. What is the output of the following script

```
typeset -i n1
typeset -i n2
n1=1
n2=1
while test $n1 -eq $n2
do
    n2=$n2+1
    print $n1
    if [ $n1 -gt $n2 ]
    then
        break
    else
        continue
    fi
    n1=$n1+1
    print $n2
done
```

output -> it will print 1 as it printed n1 only due to the continue statement.



8. Create the following menu:

- Press 1 to ls
- Press 2 to ls -a
- Press 3 to exit

Using select utility then while utility.

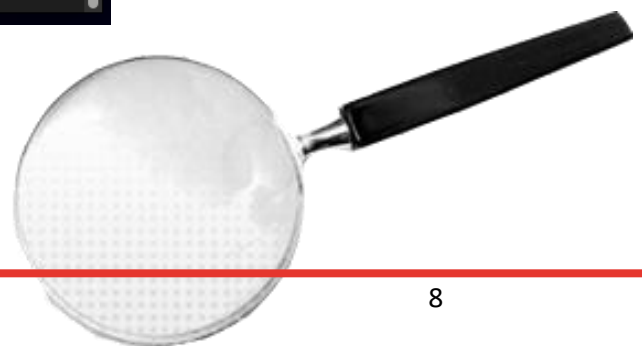
```
salma01@localhost:~ — /usr/bin/vim menu.sh
#!/bin/bash

PS3="Enter your choice: "
select choice in "ls" "ls -a" "exit"
do
case $REPLY in
1) ls
;;
2) ls -a;
;;
3) exit
;;
*) echo $REPLY is not an option
;;
esac
done

~
~
~
~
~

"menu.sh" 16L, 181B 1,1 All
```

```
salma01@localhost:~
[salma01@localhost ~]$ . menu.sh
1) ls
2) ls -a
3) exit
Enter your choice: 1
1 Documents f8 Music mycv passwd Templates
backup Downloads file1 mybackup.sh myinfo.sh Pictures testLab.sh
backup_dir email file2 mycase.sh myls.sh Public Videos
commands error hello.sh mycd.sh myteam s1.sh
Desktop f1 ls_output mychmod.sh mytest.sh s2.sh
docs f2 menu.sh mycp.sh oldpasswd scripts
Enter your choice: 2
. .cache f1 ls_output mycv .profile
.. commands f2 menu.sh myinfo.sh Public
1 .config f8 .mozilla myls.sh s1.sh
backup Desktop file1 Music myteam s2.sh
backup_dir docs file2 mybackup.sh mytest.sh scripts
.bash_history Documents .gitconfig mycase.sh oldpasswd Templates
.bash_logout Downloads hello.sh mycd.sh passwd testLab.sh
.bash_profile email .lessht mychmod.sh Pictures Videos
.bashrc error .local mycp.sh .prifile .viminfo
Enter your choice:
```



9. Write a script called myarr that ask a user how many elements he wants to enter in an array, fill the array and then print it.

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

PS3="Enter your choice: "
select choice in "ls" "ls -a" "exit"
do
case $REPLY in
    1) ls
        ;;
    2) ls -a;
        ;;
    3) exit
        ;;
    *) echo $REPLY is not an option
        ;;
esac
done
~
~
~
~
~
~
1,1 All
```

```
salma01@localhost:~
[salma01@localhost ~]$ . myarr.sh
Enter the number of elements: 3
Enter element number 1: 10
Enter element number 2: 20
Enter element number 3: 30

Entered array elements:
10
20
30
[salma01@localhost ~]$ vi menu.sh
You have new mail in /var/spool/mail/salma01
[salma01@localhost ~]$
```



10. Write a script called myavg that calculate average of all numbers entered by a user.

Note: use arrays.

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

read -p "Enter the number of elements: " num_elements

declare -a myarr

for ((i=0; i < num_elements; i++));
do
    read -p "Enter element $((i + 1)): " element
    myarr[i]=$element
done

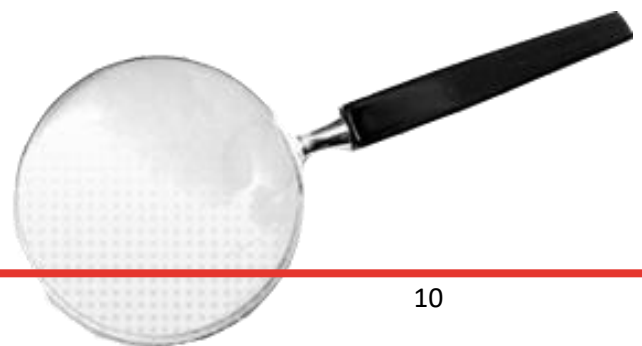
sum=0
for elements in "${myarr[@]}";
do
    sum += elements
done

average=$((sum / num_elements))

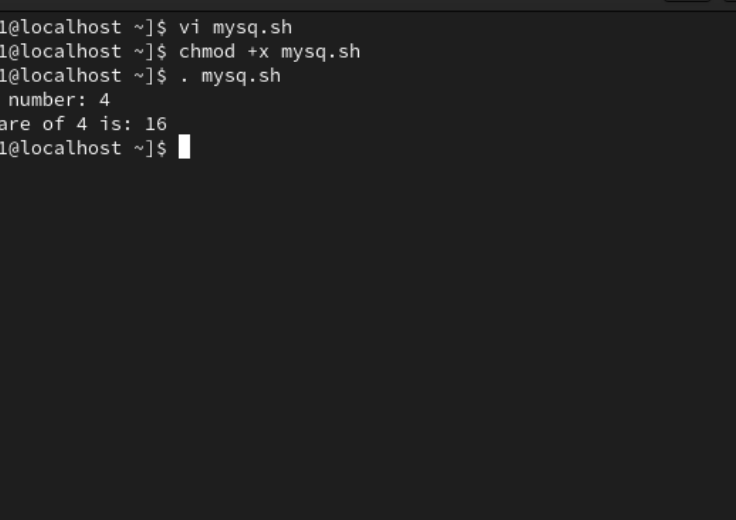
echo -e "\nEntered numbers: ${myarr[@]}"
echo "Sum of numbers: $sum"
echo "Average of numbers: $average"
"myavg.sh" 23L, 402B
```

```
salma01@localhost:~
[salma01@localhost ~]$ chmod +x myavg.sh
[salma01@localhost ~]$ . myavg.sh
Enter the number of elements: 3
Enter element 1: 10
Enter element 2: 20
Enter element 3: 30

Entered numbers: 10 20 30
Sum of numbers: 60
Average of numbers: 20
[salma01@localhost ~]$
```



11. Write a function called `mysq` that calculate square if its argument.

[illegible]

```
salma01@localhost:~  
[salma01@localhost ~]$ vi mysql.sh  
[salma01@localhost ~]$ chmod +x mysql.sh  
[salma01@localhost ~]$ . mysql.sh  
Enter a number: 4  
The square of 4 is: 16  
[salma01@localhost ~]$
```

