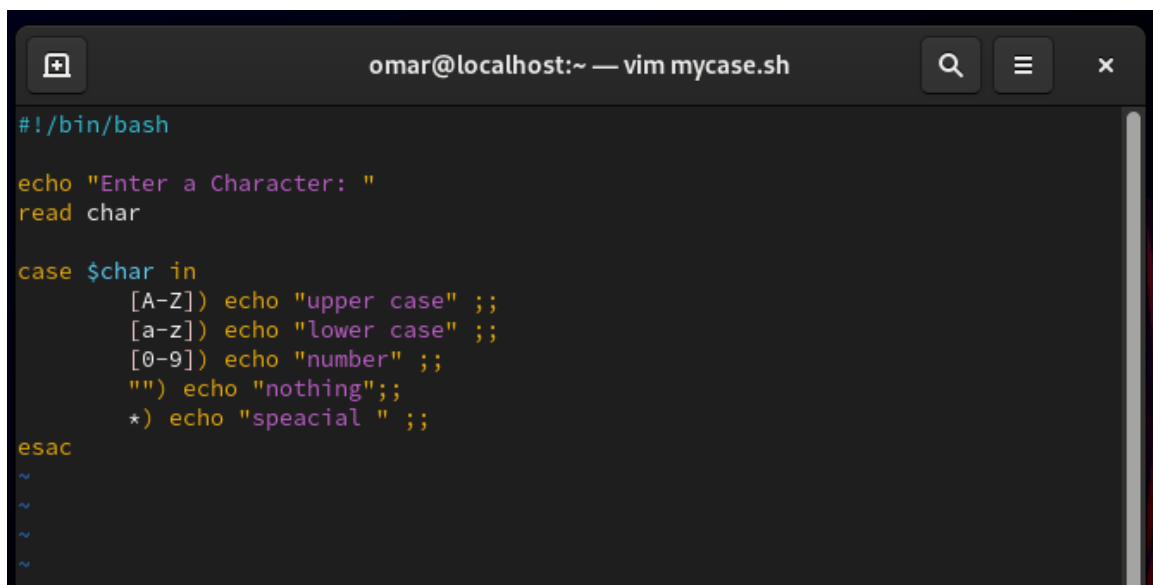


Bash lab 3

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.



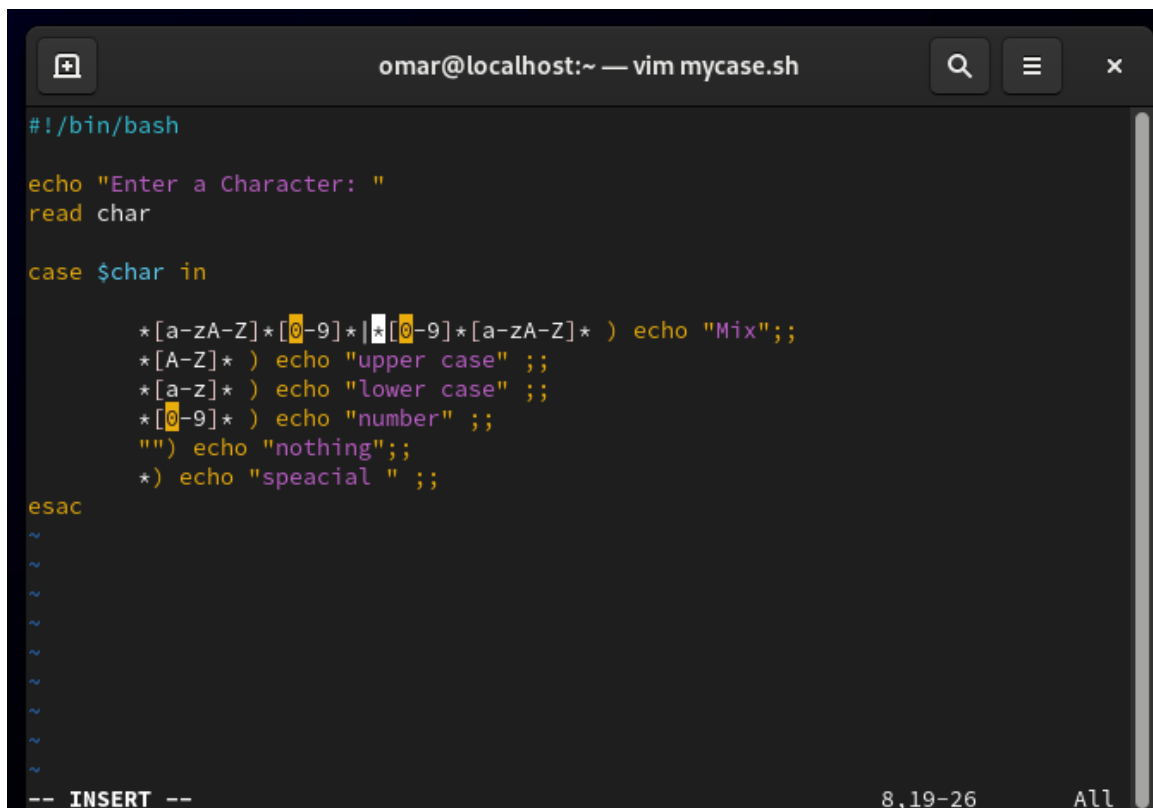
```
#!/bin/bash

echo "Enter a Character: "
read char

case $char in
    [A-Z]) echo "upper case" ;;
    [a-z]) echo "lower case" ;;
    [0-9]) echo "number" ;;
    "") echo "nothing";;
    *) echo "speacial " ;;
esac

~
~
~
~
```

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.



```
omar@localhost:~ — vim mycase.sh

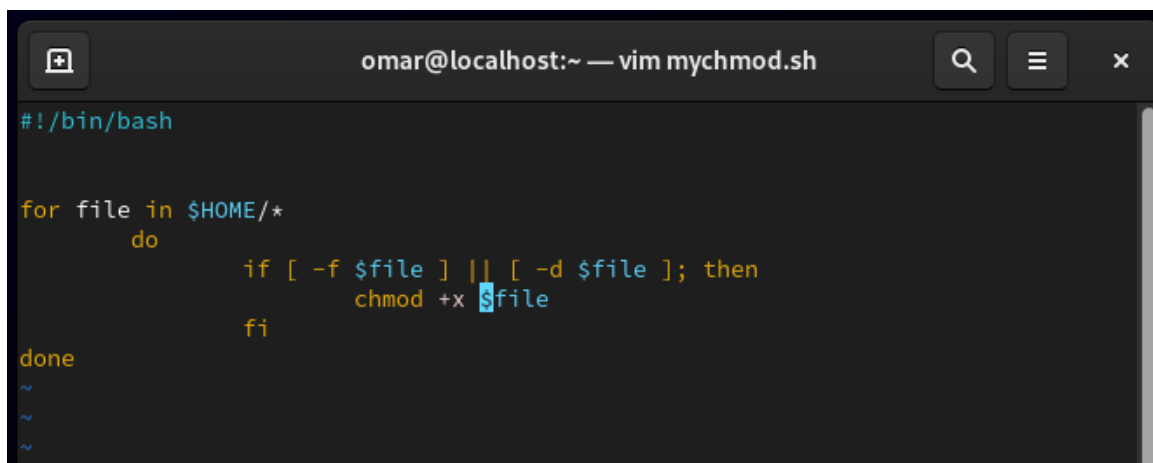
#!/bin/bash

echo "Enter a Character: "
read char

case $char in
    *[a-zA-Z]*|[0-9]*|_*[0-9]*[a-zA-Z]* ) echo "Mix";;
    *[A-Z]* ) echo "upper case" ;;
    *[a-z]* ) echo "lower case" ;;
    *[0-9]* ) echo "number" ;;
    "") echo "nothing";;
    *) echo "speacial " ;;
esac

~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

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



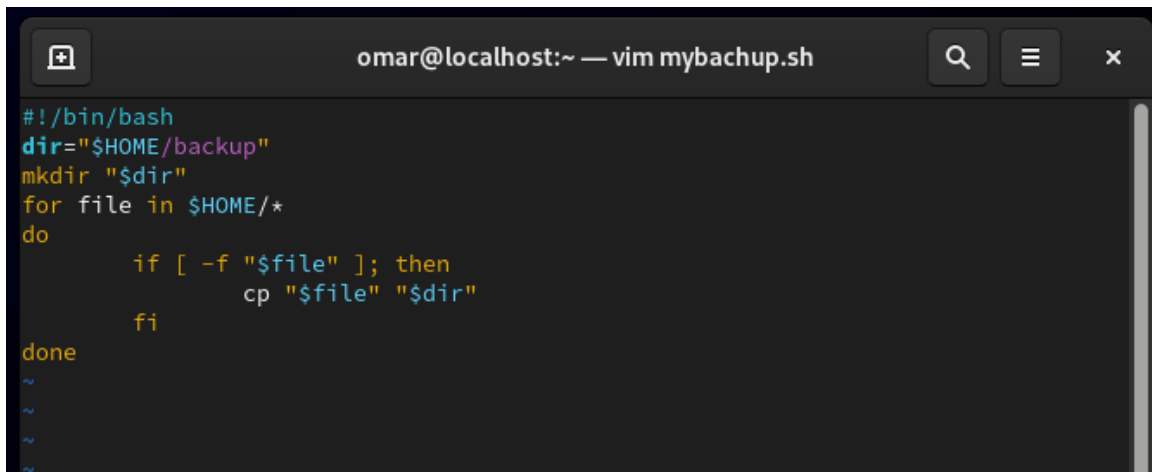
```
omar@localhost:~ — vim mychmod.sh

#!/bin/bash

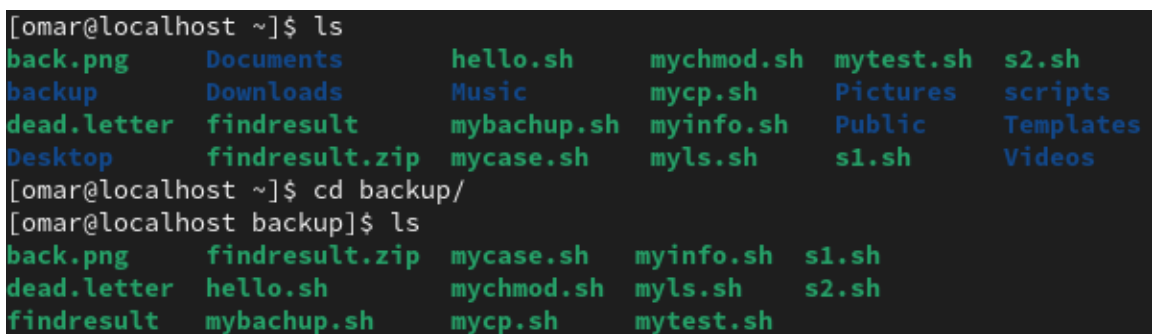
for file in $HOME/*
do
    if [ -f $file ] || [ -d $file ]; then
        chmod +x $file
    fi
done

~
~
~
```

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



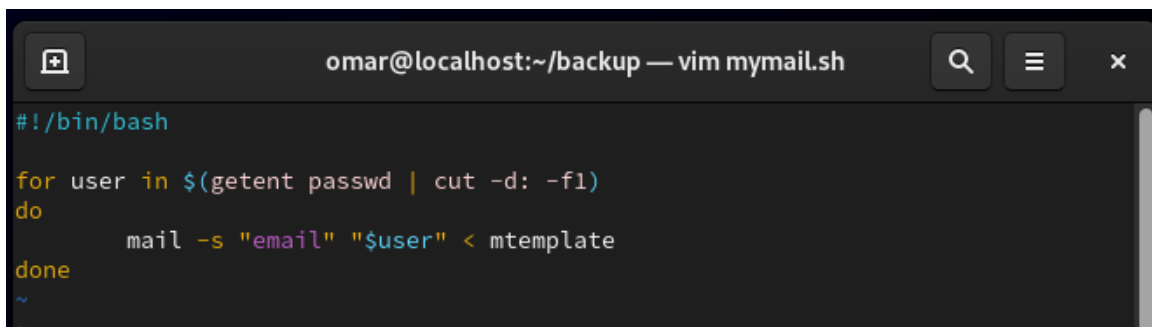
```
omar@localhost:~ — vim mybachup.sh
#!/bin/bash
dir="$HOME/backup"
mkdir "$dir"
for file in $HOME/*
do
    if [ -f "$file" ]; then
        cp "$file" "$dir"
    fi
done
~
~
~
~
```



```
[omar@localhost ~]$ ls
back.png    Documents    hello.sh    mychmod.sh  mytest.sh   s2.sh
backup      Downloads    Music       mycp.sh     Pictures     scripts
dead.letter findresult   mybachup.sh myinfo.sh   Public       Templates
Desktop     findresult.zip mycase.sh   myls.sh     s1.sh        Videos

[omar@localhost ~]$ cd backup/
[omar@localhost backup]$ ls
back.png    findresult.zip mycase.sh    myinfo.sh    s1.sh
dead.letter hello.sh        mychmod.sh   myls.sh       s2.sh
findresult  mybachup.sh    mycp.sh      mytest.sh
```

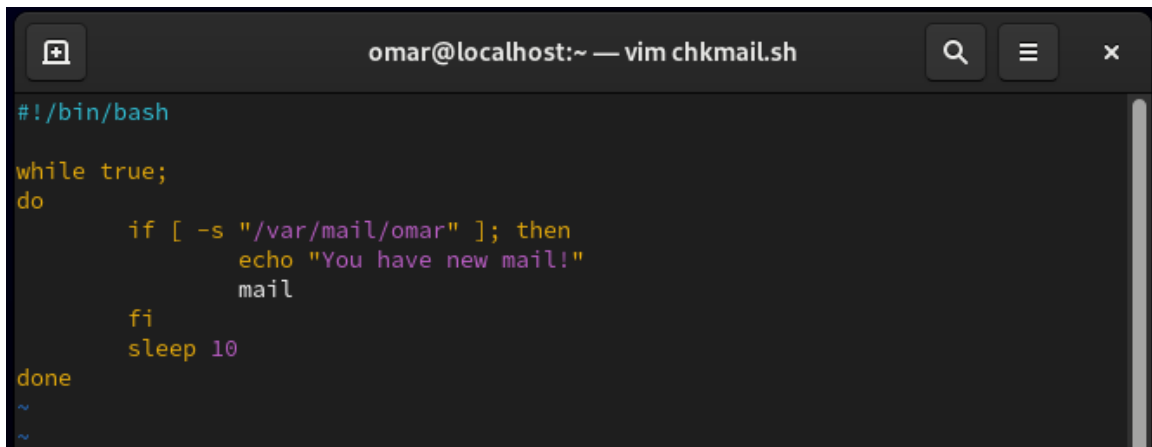
5. Write a script called mymail using for utility to send a mail to all users in the system. Note: write the mail body in a file called mtemplate.



```
omar@localhost:~/backup — vim mymail.sh
#!/bin/bash

for user in $(getent passwd | cut -d: -f1)
do
    mail -s "email" "$user" < mtemplate
done
~
~
```

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

A screenshot of a terminal window with a dark background. The title bar at the top shows 'omar@localhost:~ — vim chkmail.sh'. The terminal content is a bash script:

```
#!/bin/bash
while true;
do
    if [ -s "/var/mail/omar" ]; then
        echo "You have new mail!"
        mail
    fi
    sleep 10
done
```

Bonus: Open a talk session to a certain user when she/he logs into the system.

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
```

- Nothing will be printed because of the infinite loop

8. Create the following menu:

- a. Press 1 to ls
- b. Press 2 to ls -a
- c. Press 3 to exit

Using select utility then while utility.

```
omar@localhost:~ — vim menu.sh

#!/bin/bash

while true; do
    echo "1. Press 1 to ls"
    echo "2. Press 2 to ls -a"
    echo "3. Press 3 to exit"

    read choice

    case $choice in
        1) ls ;;
        2) ls -a;;
        3) exit 0;;
    esac
done
```

```
omar@localhost:~ — vim menu.sh

#!/bin/bash
select choice in "1. Press 1 to ls" "2.Press 2 to ls -a" "3.Press 3 to exit"; do
    case $REPLY in
        1) ls ;;
        2) ls -a;;
        3) exit 0;;
    esac
done
~
~
```

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.

```
omar@localhost:~ — vim myarr.sh

#!/bin/bash

echo "Enter array size: "
read size

typeset -i arr

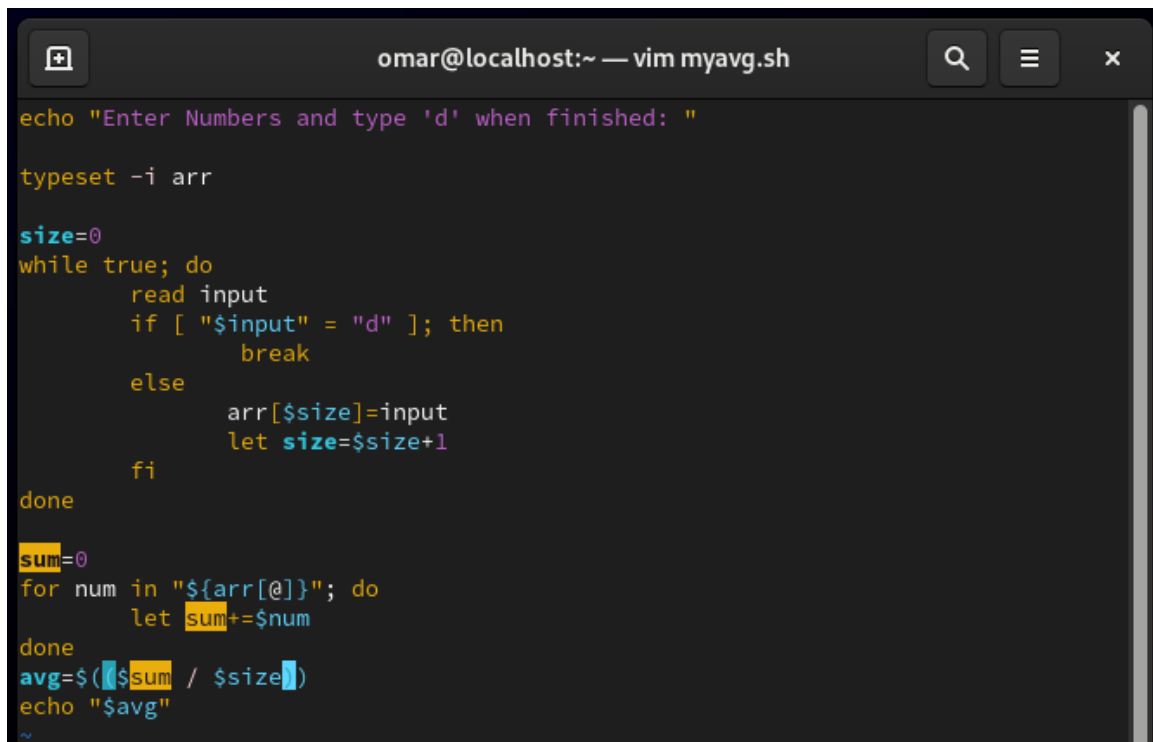
i=0

while [ $i -lt $size ]
do
    echo "enter element"
    read element
    arr[$i]=$element
    let i=$i+1
done

echo "${arr[@]}"

~
```

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



```
omar@localhost:~ — vim myavg.sh

echo "Enter Numbers and type 'd' when finished: "

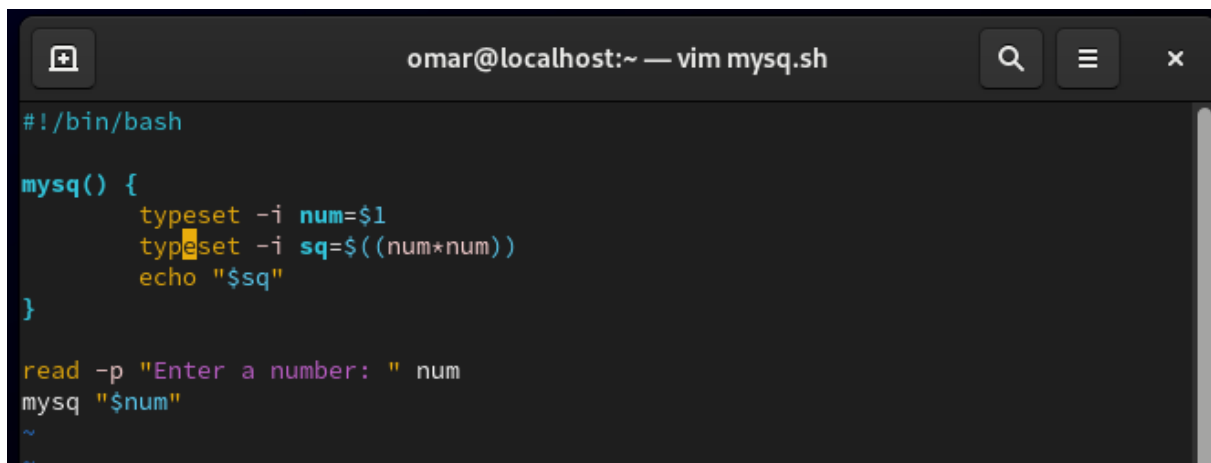
typeset -i arr

size=0
while true; do
    read input
    if [ "$input" = "d" ]; then
        break
    else
        arr[$size]=input
        let size=$size+1
    fi
done

sum=0
for num in "${arr[@]}"; do
    let sum+=num
done

avg=$((sum / size))
echo "$avg"
```

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



```
omar@localhost:~ — vim mysq.sh

#!/bin/bash

mysq() {
    typeset -i num=$1
    typeset -i sq=$((num*num))
    echo "$sq"
}

read -p "Enter a number: " num
mysq "$num"
```