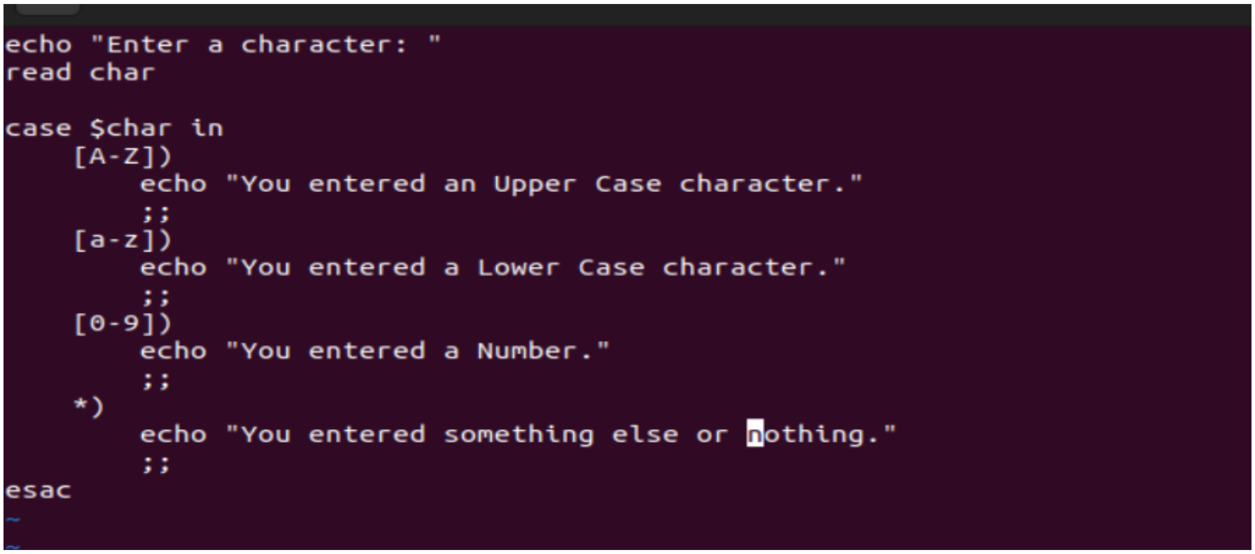
Bash Shell Script
Lab 3

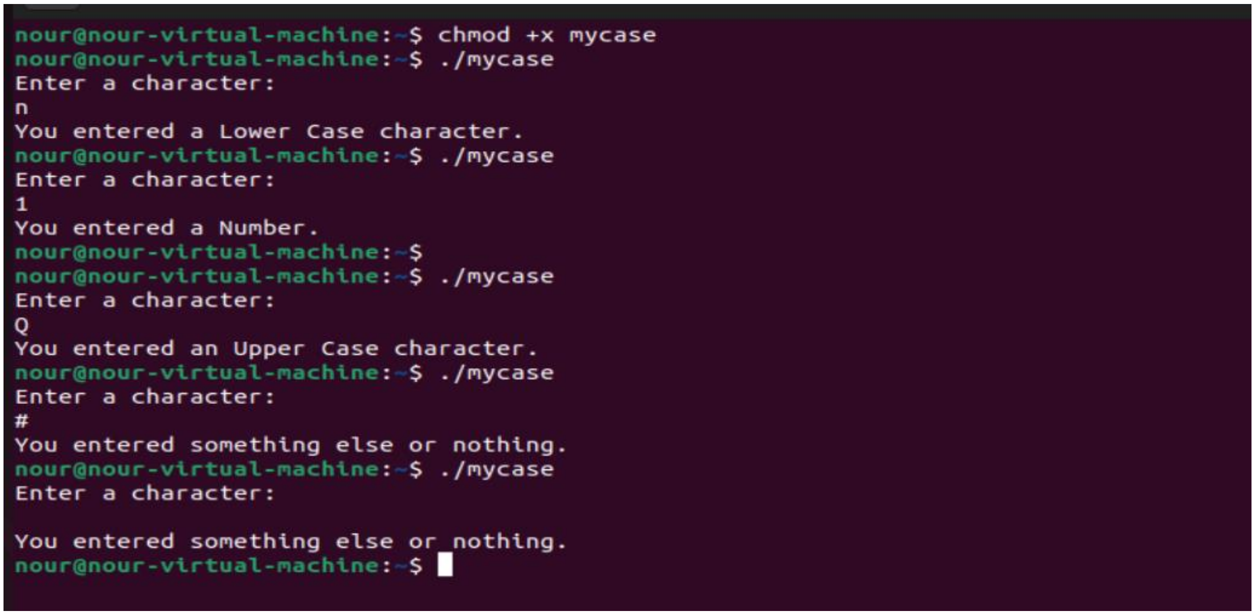
Name: Nourhan Radwan Gaber

ID: 26

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.

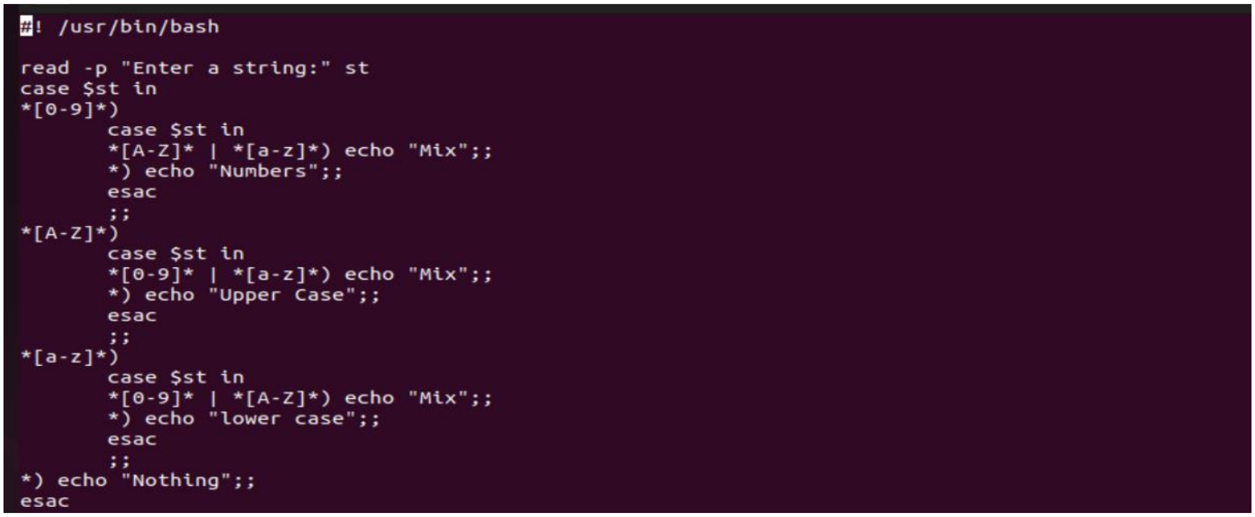


```
echo "Enter a character: "
read char

case $char in
    [A-Z])
        echo "You entered an Upper Case character."
        ;;
    [a-z])
        echo "You entered a Lower Case character."
        ;;
    [0-9])
        echo "You entered a Number."
        ;;
    *)
        echo "You entered something else or nothing."
        ;;
esac
~
~
```

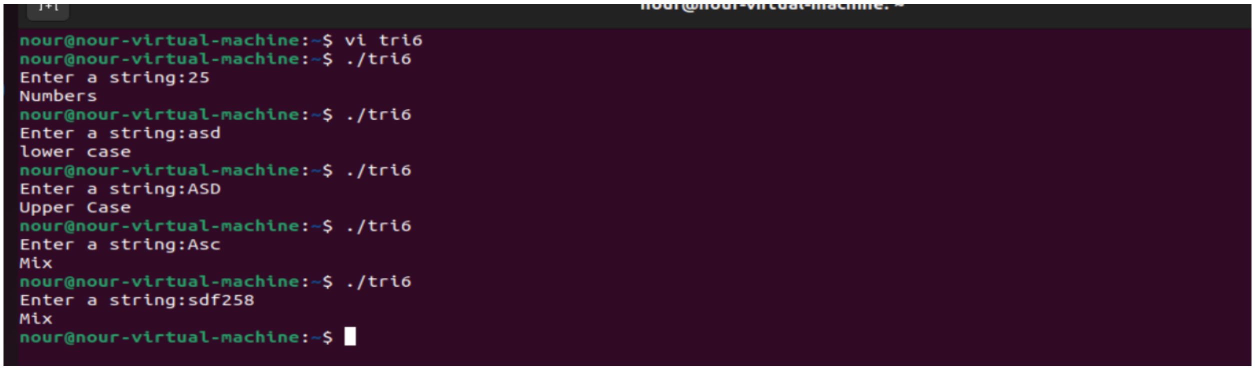


```
nour@nour-virtual-machine:~$ chmod +x mycase
nour@nour-virtual-machine:~$ ./mycase
Enter a character:
n
You entered a Lower Case character.
nour@nour-virtual-machine:~$ ./mycase
Enter a character:
1
You entered a Number.
nour@nour-virtual-machine:~$
nour@nour-virtual-machine:~$ ./mycase
Enter a character:
Q
You entered an Upper Case character.
nour@nour-virtual-machine:~$ ./mycase
Enter a character:
#
You entered something else or nothing.
nour@nour-virtual-machine:~$ ./mycase
Enter a character:

You entered something else or nothing.
nour@nour-virtual-machine:~$
```

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.

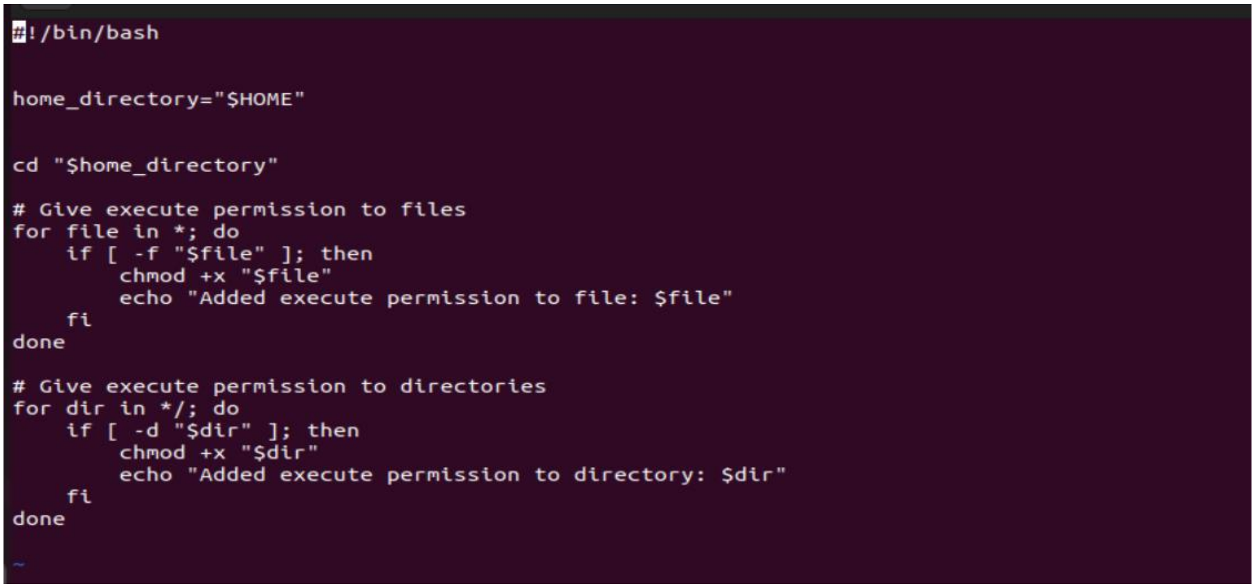


```
#! /usr/bin/bash

read -p "Enter a string:" st
case $st in
*[0-9]*)
       case $st in
       *[A-Z]* | *[a-z]*) echo "Mix";;
       *) echo "Numbers";;
       esac
       ;;
*[A-Z]*)
       case $st in
       *[0-9]* | *[a-z]*) echo "Mix";;
       *) echo "Upper Case";;
       esac
       ;;
*[a-z]*)
       case $st in
       *[0-9]* | *[A-Z]*) echo "Mix";;
       *) echo "lower case";;
       esac
       ;;
*) echo "Nothing";;
esac
```

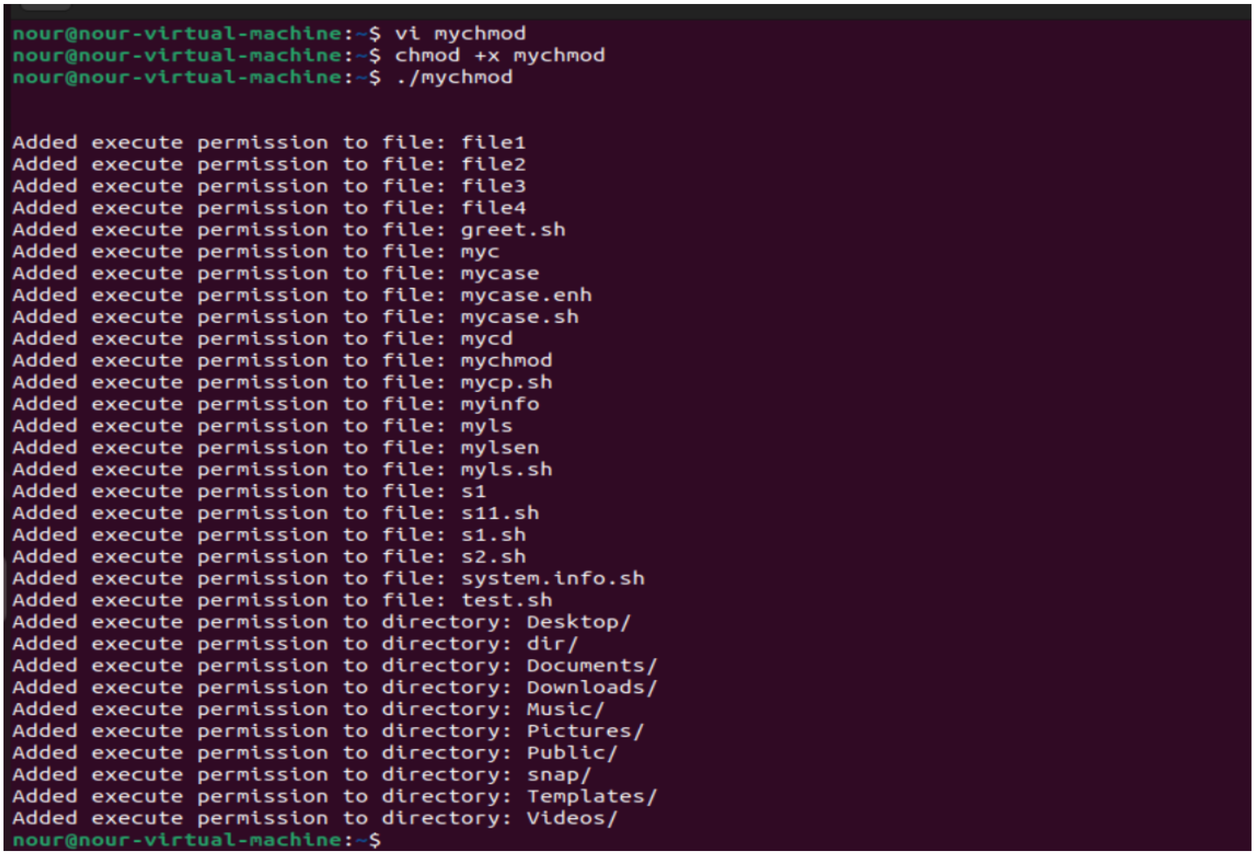


```
nour@nour-virtual-machine:~$ vi tri6
nour@nour-virtual-machine:~$ ./tri6
Enter a string:25
Numbers
nour@nour-virtual-machine:~$ ./tri6
Enter a string:asd
lower case
nour@nour-virtual-machine:~$ ./tri6
Enter a string:ASD
Upper Case
nour@nour-virtual-machine:~$ ./tri6
Enter a string:Asc
Mix
nour@nour-virtual-machine:~$ ./tri6
Enter a string:sdf258
Mix
nour@nour-virtual-machine:~$ 
```

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

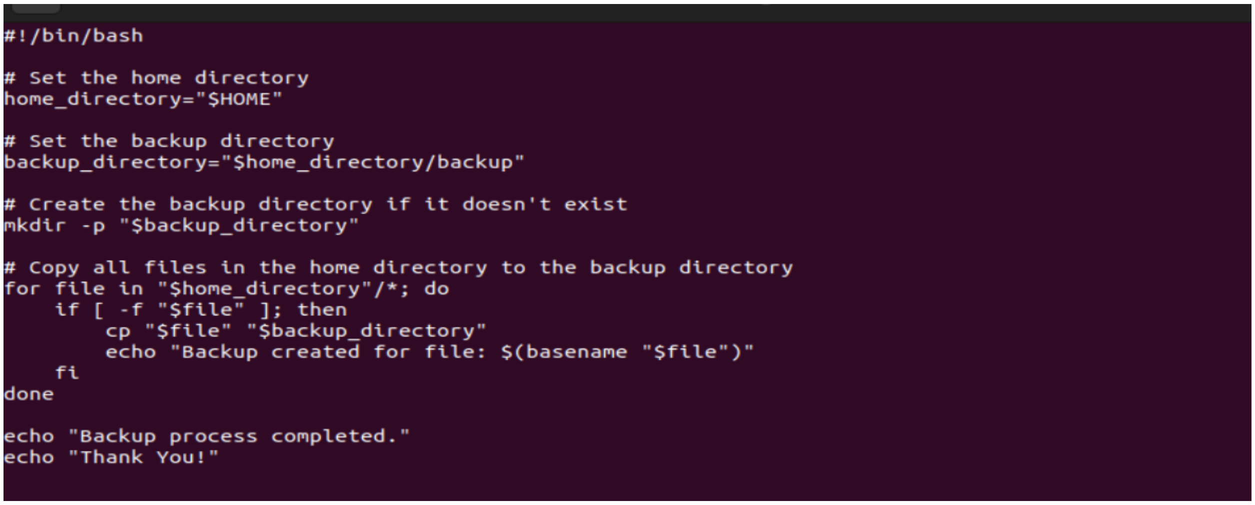


```
#!/bin/bash


home_directory="$HOME"


cd "$home_directory"

# Give execute permission to files
for file in *; do
    if [ -f "$file" ]; then
        chmod +x "$file"
        echo "Added execute permission to file: $file"
    fi
done

# Give execute permission to directories
for dir in */; do
    if [ -d "$dir" ]; then
        chmod +x "$dir"
        echo "Added execute permission to directory: $dir"
    fi
done

~
```

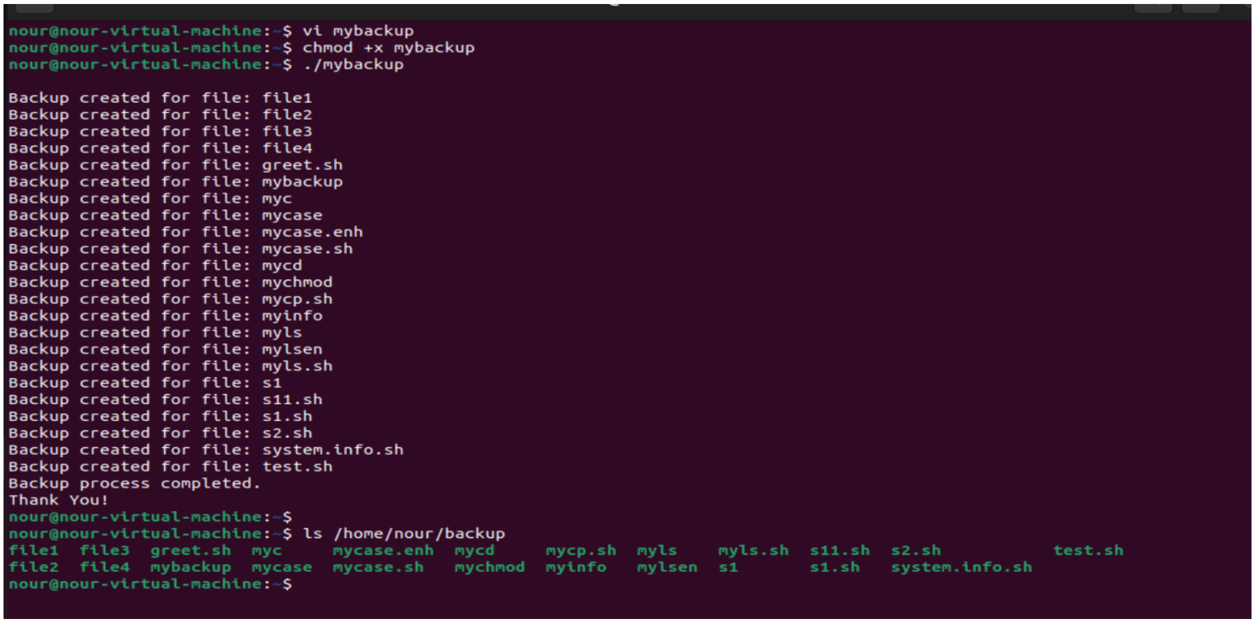


```
nour@nour-virtual-machine:~$ vi mychmod
nour@nour-virtual-machine:~$ chmod +x mychmod
nour@nour-virtual-machine:~$ ./mychmod


Added execute permission to file: file1
Added execute permission to file: file2
Added execute permission to file: file3
Added execute permission to file: file4
Added execute permission to file: greet.sh
Added execute permission to file: myc
Added execute permission to file: mycase
Added execute permission to file: mycase.enh
Added execute permission to file: mycase.sh
Added execute permission to file: mycd
Added execute permission to file: mychmod
Added execute permission to file: mycp.sh
Added execute permission to file: myinfo
Added execute permission to file: myls
Added execute permission to file: mylsen
Added execute permission to file: myls.sh
Added execute permission to file: s1
Added execute permission to file: s11.sh
Added execute permission to file: s1.sh
Added execute permission to file: s2.sh
Added execute permission to file: system.info.sh
Added execute permission to file: test.sh
Added execute permission to directory: Desktop/
Added execute permission to directory: dir/
Added execute permission to directory: Documents/
Added execute permission to directory: Downloads/
Added execute permission to directory: Music/
Added execute permission to directory: Pictures/
Added execute permission to directory: Public/
Added execute permission to directory: snap/
Added execute permission to directory: Templates/
Added execute permission to directory: Videos/
nour@nour-virtual-machine:~$
```

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

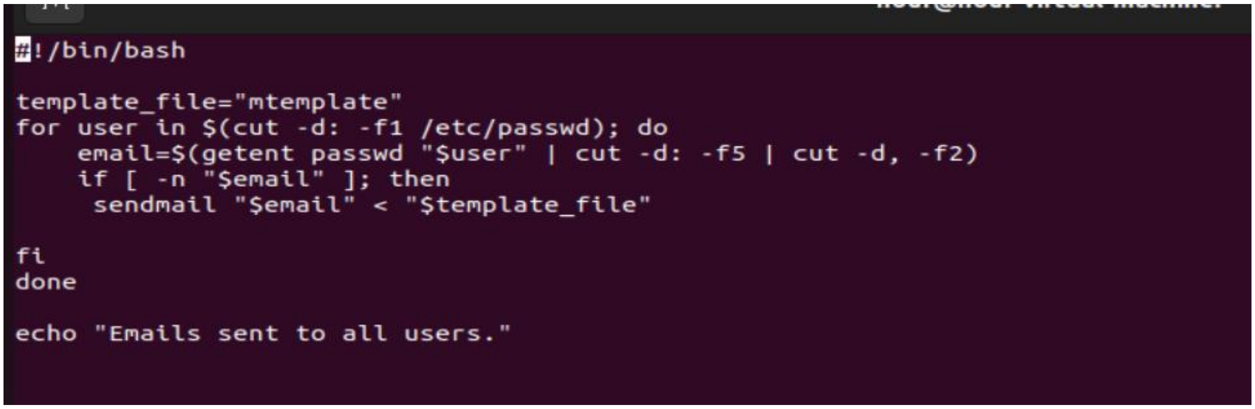


```
#!/bin/bash

# Set the home directory
home_directory="$HOME"

# Set the backup directory
backup_directory="$home_directory/backup"

# Create the backup directory if it doesn't exist
mkdir -p "$backup_directory"

# Copy all files in the home directory to the backup directory
for file in "$home_directory"/*; do
    if [ -f "$file" ]; then
        cp "$file" "$backup_directory"
        echo "Backup created for file: $(basename "$file")"
    fi
done

echo "Backup process completed."
echo "Thank You!"
```

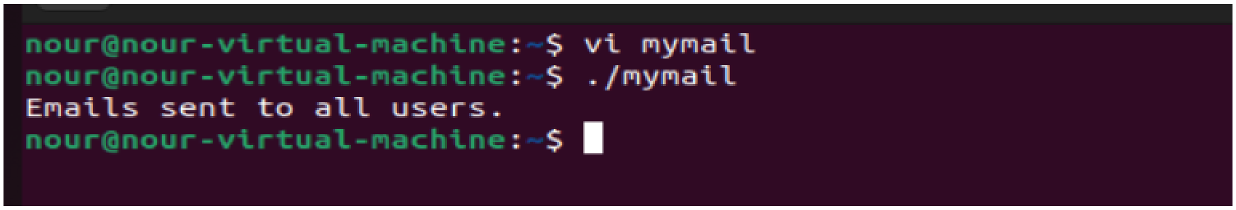


```
nour@nour-virtual-machine:~$ vi mybackup
nour@nour-virtual-machine:~$ chmod +x mybackup
nour@nour-virtual-machine:~$ ./mybackup

Backup created for file: file1
Backup created for file: file2
Backup created for file: file3
Backup created for file: file4
Backup created for file: greet.sh
Backup created for file: mybackup
Backup created for file: myc
Backup created for file: mycase
Backup created for file: mycase.enh
Backup created for file: mycase.sh
Backup created for file: mycd
Backup created for file: mychmod
Backup created for file: mycp.sh
Backup created for file: myinfo
Backup created for file: myls
Backup created for file: mylsen
Backup created for file: myls.sh
Backup created for file: s1
Backup created for file: s11.sh
Backup created for file: s1.sh
Backup created for file: s2.sh
Backup created for file: system.info.sh
Backup created for file: test.sh
Backup process completed.
Thank You!
nour@nour-virtual-machine:~$
nour@nour-virtual-machine:~$ ls /home/nour/backup
file1  file3  greet.sh  myc     mycase.enh  mycd     mycp.sh  myls    myls.sh  s11.sh  s2.sh           test.sh
file2  file4  mybackup  mycase  mycase.sh   mychmod  myinfo   mylsen  s1       s1.sh   system.info.sh
nour@nour-virtual-machine:~$
```

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.

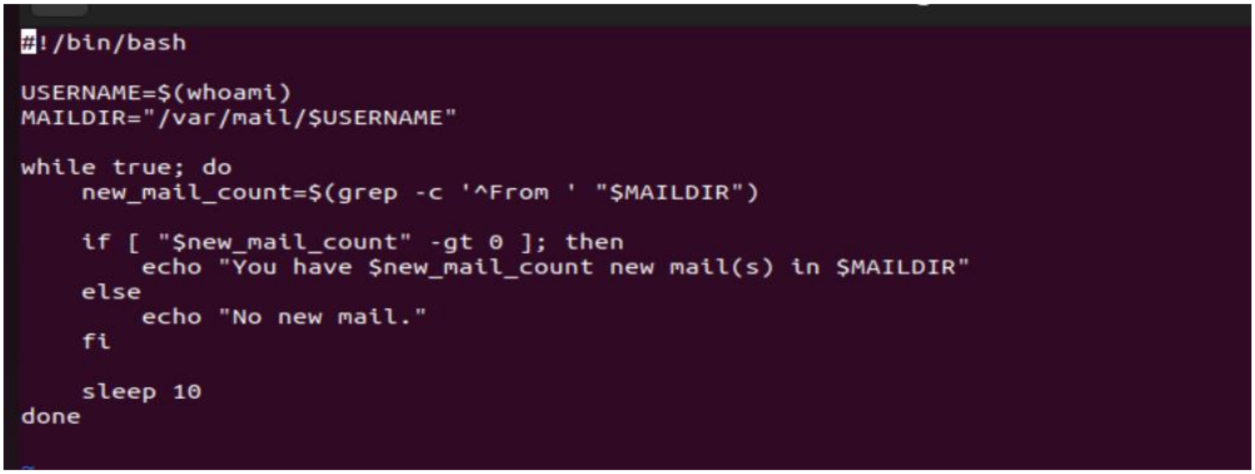


```
#!/bin/bash

template_file="mtemplate"
for user in $(cut -d: -f1 /etc/passwd); do
    email=$(getent passwd "$user" | cut -d: -f5 | cut -d, -f2)
    if [ -n "$email" ]; then
     sendmail "$email" < "$template_file"

fi
done

echo "Emails sent to all users."
```

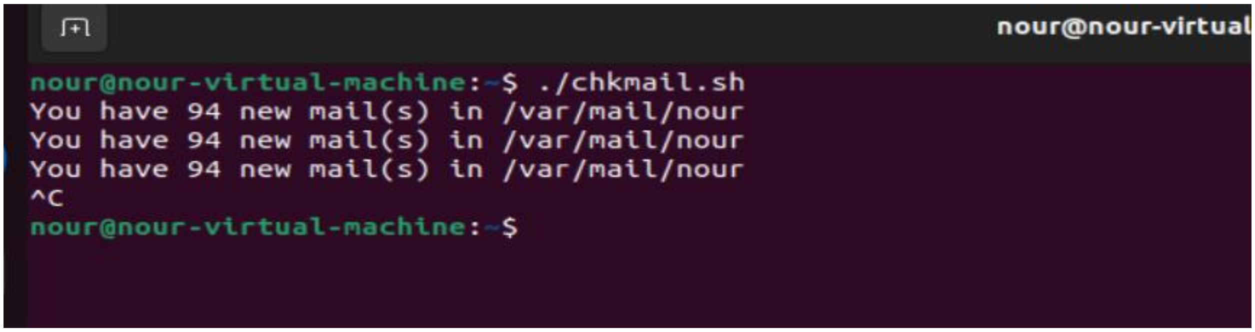


```
nour@nour-virtual-machine:~$ vi mymail
nour@nour-virtual-machine:~$ ./mymail
Emails sent to all users.
nour@nour-virtual-machine:~$ 
```

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

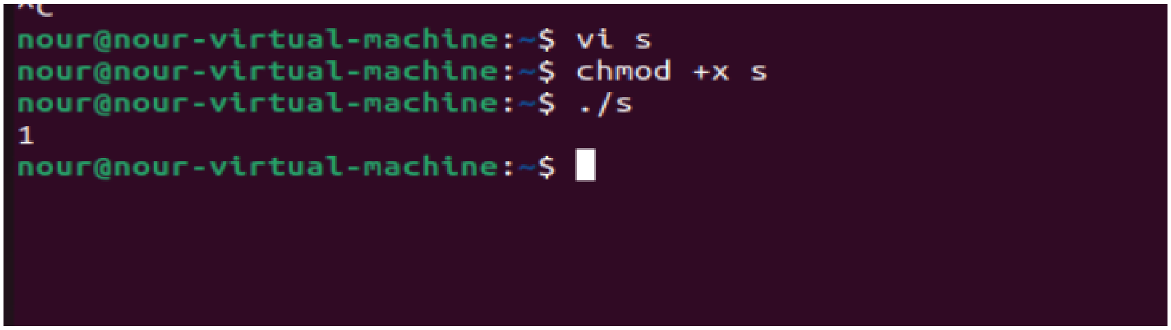


```
#!/bin/bash

USERNAME=$(whoami)
MAILDIR="/var/mail/$USERNAME"

while true; do
    new_mail_count=$(grep -c '^From ' "$MAILDIR")

    if [ "$new_mail_count" -gt 0 ]; then
        echo "You have $new_mail_count new mail(s) in $MAILDIR"
    else
        echo "No new mail."
    fi

    sleep 10
done
```



```
nour@nour-virtual-machine:~$ ./chkmail.sh
You have 94 new mail(s) in /var/mail/nour
You have 94 new mail(s) in /var/mail/nour
You have 94 new mail(s) in /var/mail/nour
^C
nour@nour-virtual-machine:~$
```

7. What is the output of the following script



```
^C
nour@nour-virtual-machine:~$ vi s
nour@nour-virtual-machine:~$ chmod +x s
nour@nour-virtual-machine:~$ ./s
1
nour@nour-virtual-machine:~$ 
```

I think, this is because the loop condition is while [$n1 -eq $n2] so, both n1 and n2 are set to 1.

The loop will run once, printing 1, and then exit the loop because the condition $n1 -eq $n2 is no longer true after incrementing n2.

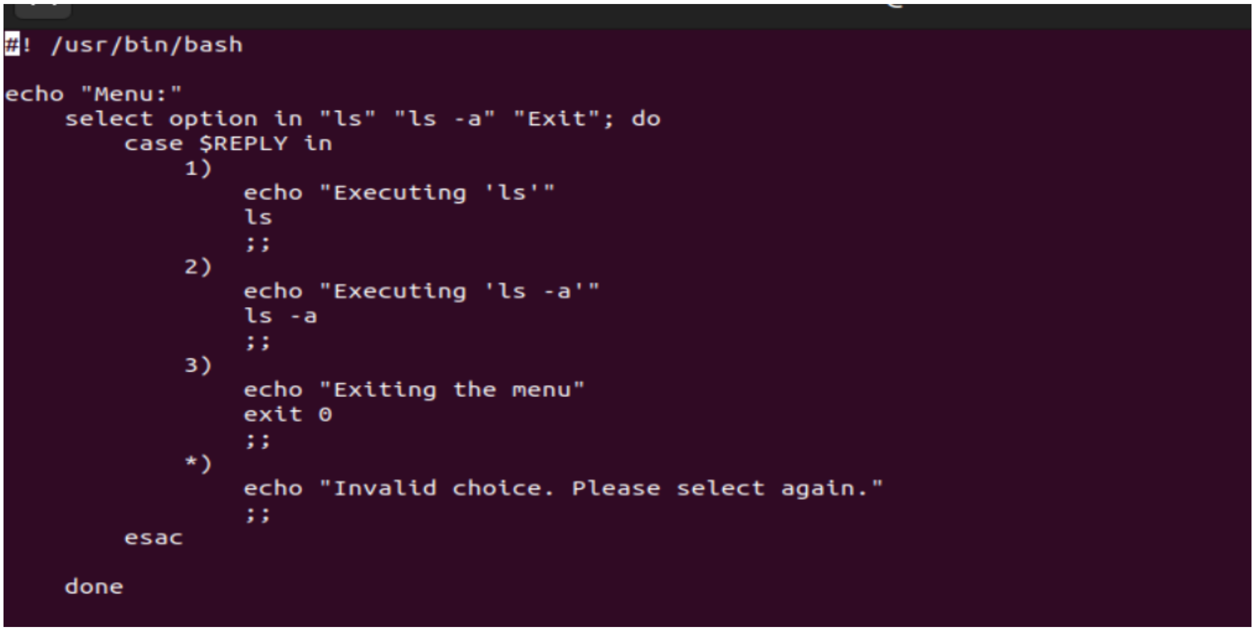
8. Create the following menu:

a. Press 1 to ls

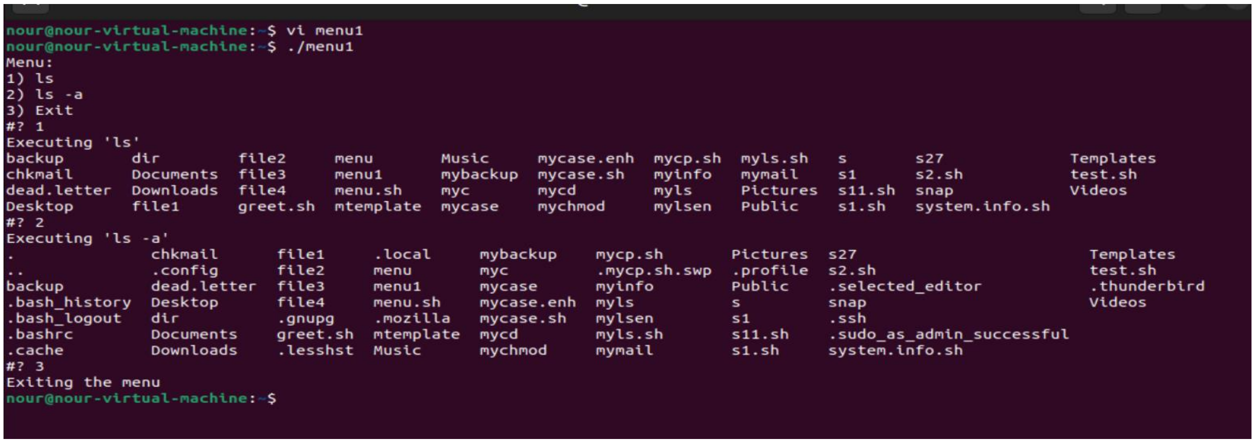
b. Press 2 to ls –a

c. Press 3 to exit

Using select utility:

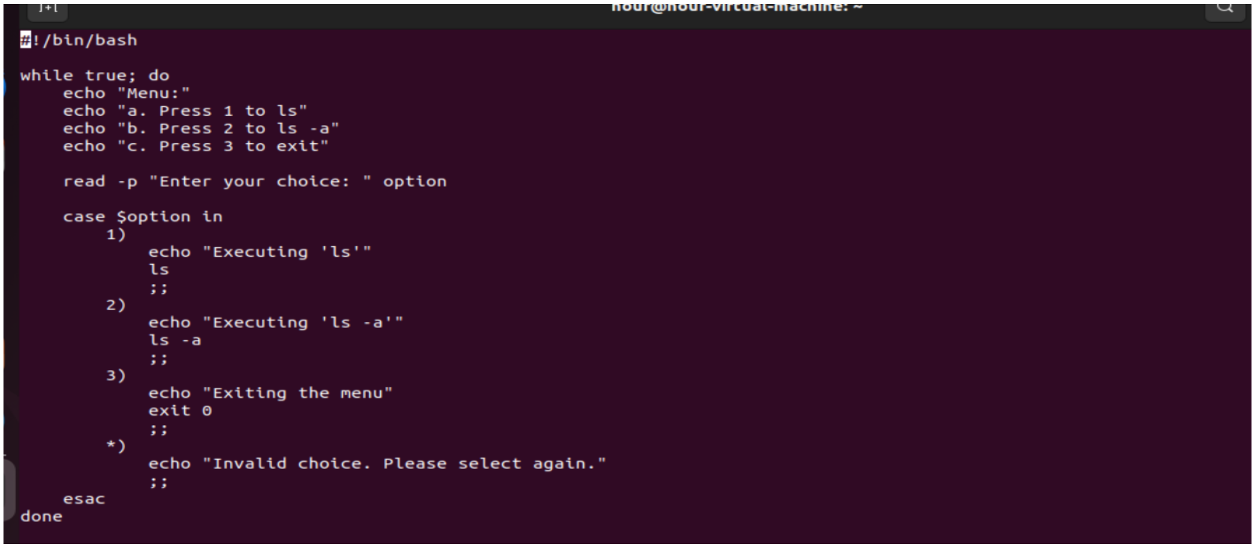


```
#! /usr/bin/bash

echo "Menu:"
    select option in "ls" "ls -a" "Exit"; do
        case $REPLY in
            1)
                echo "Executing 'ls'"
                ls
                ;;
            2)
                echo "Executing 'ls -a'"
                ls -a
                ;;
            3)
                echo "Exiting the menu"
                exit 0
                ;;
            *)
                echo "Invalid choice. Please select again."
                ;;
        esac

    done
```

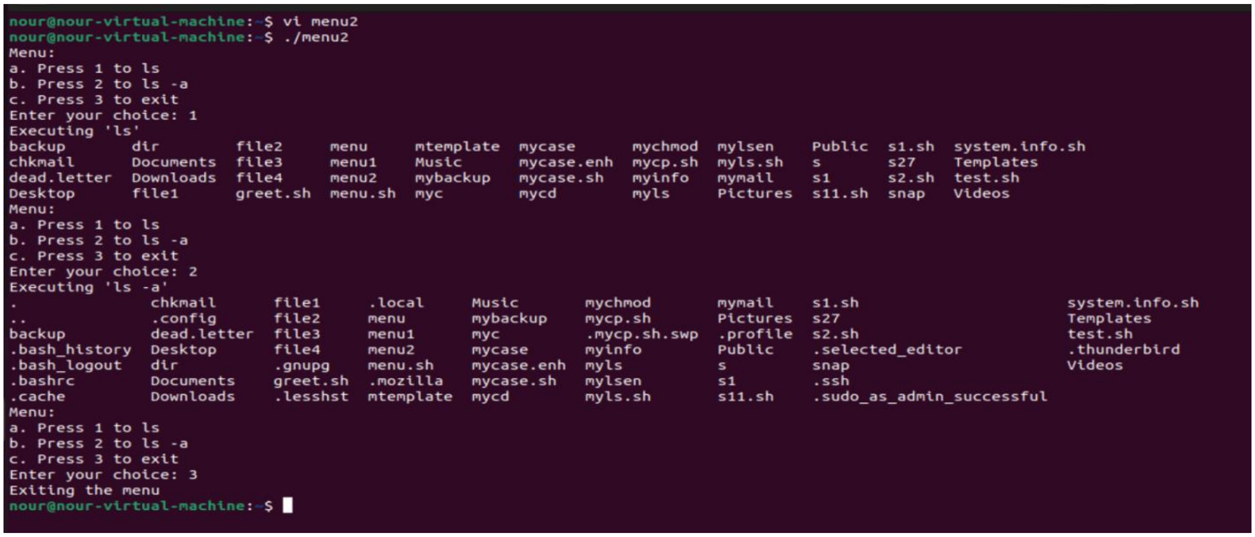


```
nour@nour-virtual-machine:~$ vi menu1
nour@nour-virtual-machine:~$ ./menu1
Menu:
1) ls
2) ls -a
3) Exit
#? 1
Executing 'ls'
backup       dir        file2     menu       Music     mycase.enh  mycp.sh  myls.sh   s       s27             Templates
chkmail      Documents  file3     menu1      mybackup  mycase.sh   myinfo   mymail    s1      s2.sh           test.sh
dead.letter  Downloads  file4     menu.sh    myc       mycd        myls     Pictures  s11.sh  snap            Videos
Desktop      file1      greet.sh  mtemplate  mycase    mychmod     mylsen   Public    s1.sh   system.info.sh
#? 2
Executing 'ls -a'
.              chkmail      file1     .local     mybackup    mycp.sh       Pictures  s27                        Templates
..             .config      file2     menu       myc         .mycp.sh.swp  .profile  s2.sh                      test.sh
backup         dead.letter  file3     menu1      mycase      myinfo        Public    .selected_editor           .thunderbird
.bash_history  Desktop      file4     menu.sh    mycase.enh  myls          s         snap                       Videos
.bash_logout   dir          .gnupg    .mozilla   mycase.sh   mylsen        s1        .ssh
.bashrc        Documents    greet.sh  mtemplate  mycd        myls.sh       s11.sh    .sudo_as_admin_successful
.cache         Downloads    .lesshst  Music      mychmod     mymail        s1.sh     system.info.sh
#? 3
Exiting the menu
nour@nour-virtual-machine:~$
```

Then while utility:

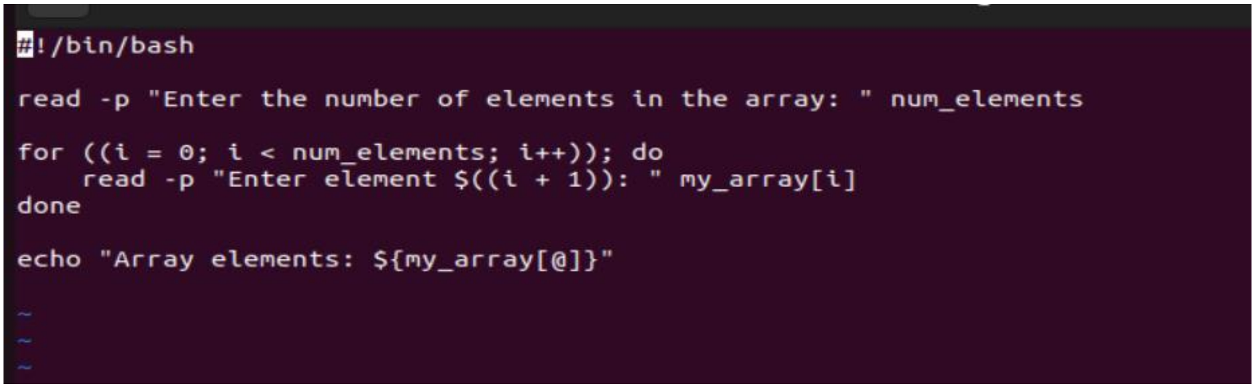


```
#!/bin/bash

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

    read -p "Enter your choice: " option

    case $option in
        1)
            echo "Executing 'ls'"
            ls
            ;;
        2)
            echo "Executing 'ls -a'"
            ls -a
            ;;
        3)
            echo "Exiting the menu"
            exit 0
            ;;
        *)
            echo "Invalid choice. Please select again."
            ;;
    esac
done
```

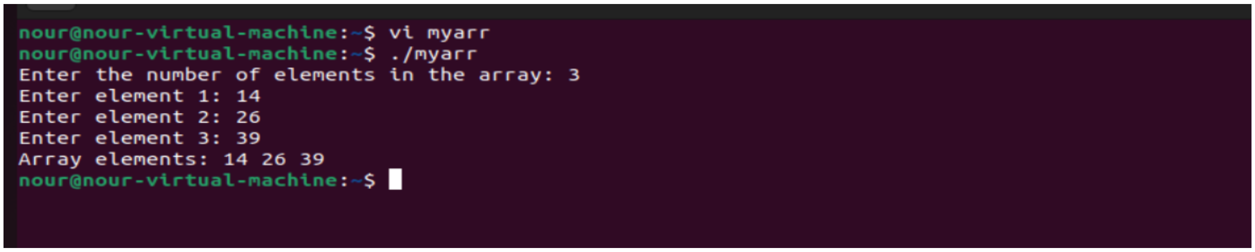


```
nour@nour-virtual-machine:~$ vi menu2
nour@nour-virtual-machine:~$ ./menu2
Menu:
a. Press 1 to ls
b. Press 2 to ls -a
c. Press 3 to exit
Enter your choice: 1
Executing 'ls'
backup        dir        file2     menu     mtemplate  mycase      mychmod  mylsen    Public  s1.sh  system.info.sh
chkmail       Documents  file3     menu1    Music      mycase.enh  mycp.sh  myls.sh   s       s27    Templates
dead.letter   Downloads  file4     menu2    mybackup   mycase.sh   myinfo   mymail    s1      s2.sh  test.sh
Desktop       file1      greet.sh  menu.sh  myc        mycd        myls     Pictures  s11.sh  snap   Videos
Menu:
a. Press 1 to ls
b. Press 2 to ls -a
c. Press 3 to exit
Enter your choice: 2
Executing 'ls -a'
.              chkmail      file1     .local     Music       mychmod       mymail    s1.sh                      system.info.sh
..             .config      file2     menu       mybackup    mycp.sh       Pictures  s27                        Templates
backup         dead.letter  file3     menu1      myc         .mycp.sh.swp  .profile  s2.sh                      test.sh
.bash_history  Desktop      file4     menu2      mycase      myinfo        Public    .selected_editor           .thunderbird
.bash_logout   dir          .gnupg    menu.sh    mycase.enh  myls          s         snap                       Videos
.bashrc        Documents    greet.sh  .mozilla   mycase.sh   mylsen        s1        .ssh
.cache         Downloads    .lesshst  mtemplate  mycd        myls.sh       s11.sh    .sudo_as_admin_successful
Menu:
a. Press 1 to ls
b. Press 2 to ls -a
c. Press 3 to exit
Enter your choice: 3
Exiting the menu
nour@nour-virtual-machine:~$ 
```

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.

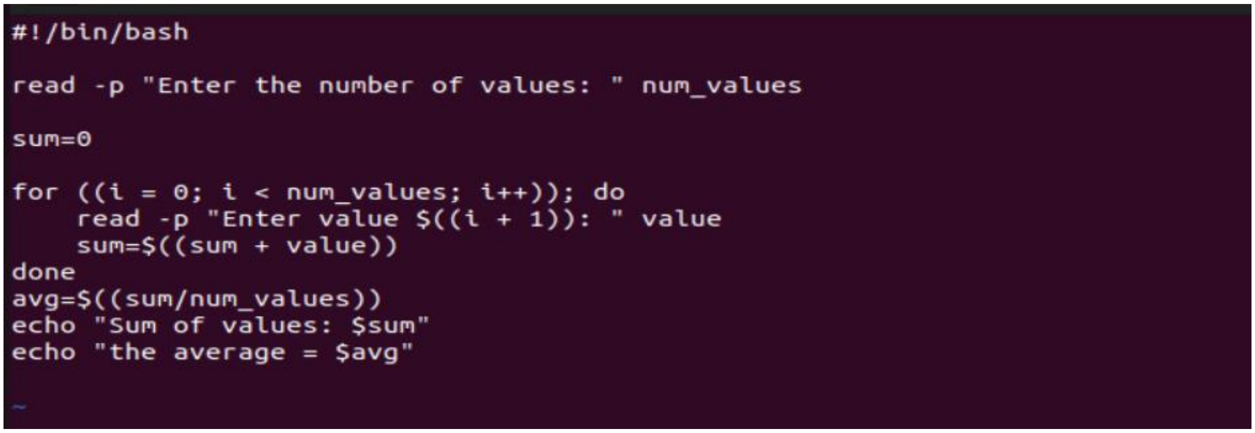


```
#!/bin/bash

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

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

echo "Array elements: ${my_array[@]}"

~
~
~
```

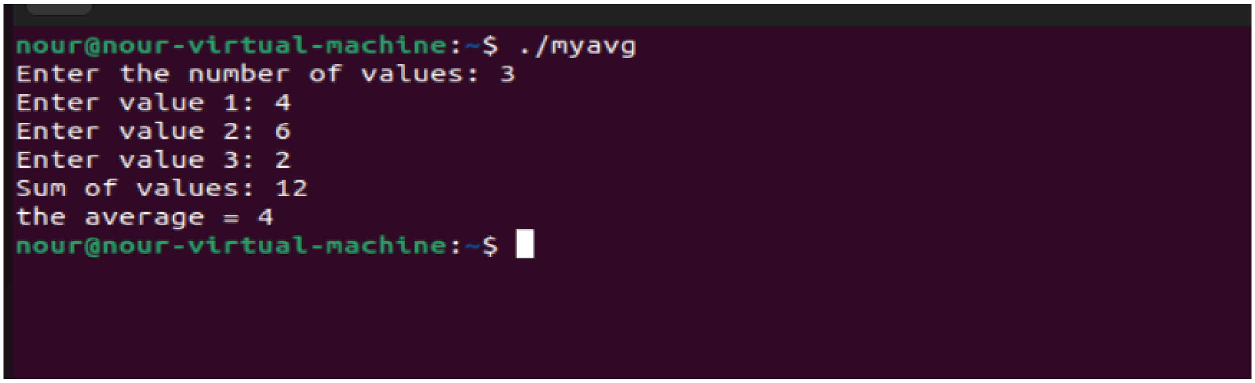


```
nour@nour-virtual-machine:~$ vi myarr
nour@nour-virtual-machine:~$ ./myarr
Enter the number of elements in the array: 3
Enter element 1: 14
Enter element 2: 26
Enter element 3: 39
Array elements: 14 26 39
nour@nour-virtual-machine:~$ 
```

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

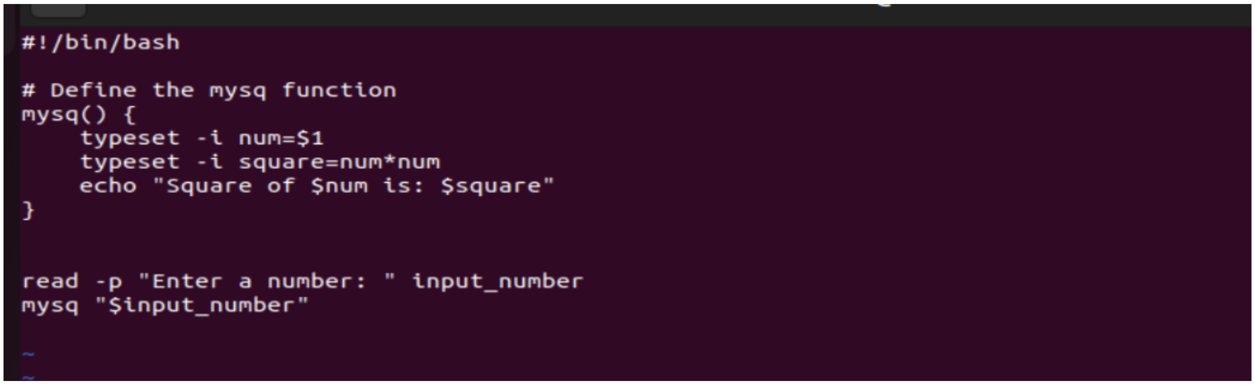


```
#!/bin/bash

read -p "Enter the number of values: " num_values

sum=0

for ((i = 0; i < num_values; i++)); do
    read -p "Enter value $((i + 1)): " value
    sum=$((sum + value))
done
avg=$((sum/num_values))
echo "Sum of values: $sum"
echo "the average = $avg"

~
```

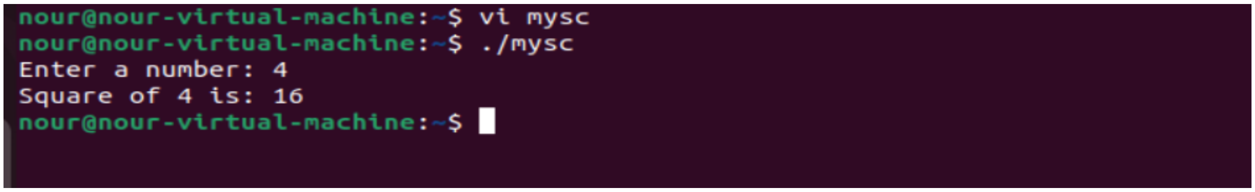


```
nour@nour-virtual-machine:~$ ./myavg
Enter the number of values: 3
Enter value 1: 4
Enter value 2: 6
Enter value 3: 2
Sum of values: 12
the average = 4
nour@nour-virtual-machine:~$ 
```

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



```
#!/bin/bash

# Define the mysq function
mysq() {
    typeset -i num=$1
    typeset -i square=num*num
    echo "Square of $num is: $square"
}


read -p "Enter a number: " input_number
mysq "$input_number"

~
~
```



```
nour@nour-virtual-machine:~$ vi mysc
nour@nour-virtual-machine:~$ ./mysc
Enter a number: 4
Square of 4 is: 16
nour@nour-virtual-machine:~$ 
```