

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.

```
touch mycase.sh
```

```
#!/usr/bin/bash  
shopt -s extglob
```

```
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 "Invalid input."  
    ;;  
esac
```

```
# Terminal =>  
# ./mycase.sh  
# M  
# m  
# 9  
#  
# -
```

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
M
Upper Case.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
m
Lower Case.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
1
Number.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
Nothing.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
-
Invalid input.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase.sh
Enter a Character:
Md
Invalid input.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$
```

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

```
touch mycase2.sh
```

```
#!/usr/bin/bash
```

```
shopt -s extglob
```

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

```
case $char in  
    "")  
        echo "Nothing."  
        ;;  
  
    +([A-Z]|" "))  
        echo "Upper Case."  
        ;;  
  
    +([a-z]|" ")  
        echo "Lower Case."  
        ;;  
  
    +([0-9]|" ")  
        echo "Number."  
        ;;  
  
    +([a-zA-Z0-9]|" ")  
        echo "Mix."  
        ;;  
  
    *)  
        echo "Invalid input."  
        ;;  
esac
```

```
# Terminal =>  
# ./mycase2.sh  
# Mazen  
# mazen  
# 20
```

Mazen Saad
mAzEn20

-

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
M
Upper Case.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
m
Lower Case.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
1
Number.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
Ma
Mix.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
Mazen Saad
Mix.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
M3s
Mix.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:

Nothing.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mycase2.sh
Enter a Character:
-
Invalid input.
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$
```

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

```
touch mychmod.sh
```

```
cd ~/Documents/lab3
```

```
mkdir ma mz me
```

```
touch file{0..9}.txt
```

```
#!/usr/bin/bash
```

```
shopt -s extglob
```

```
for item in $1/*
```

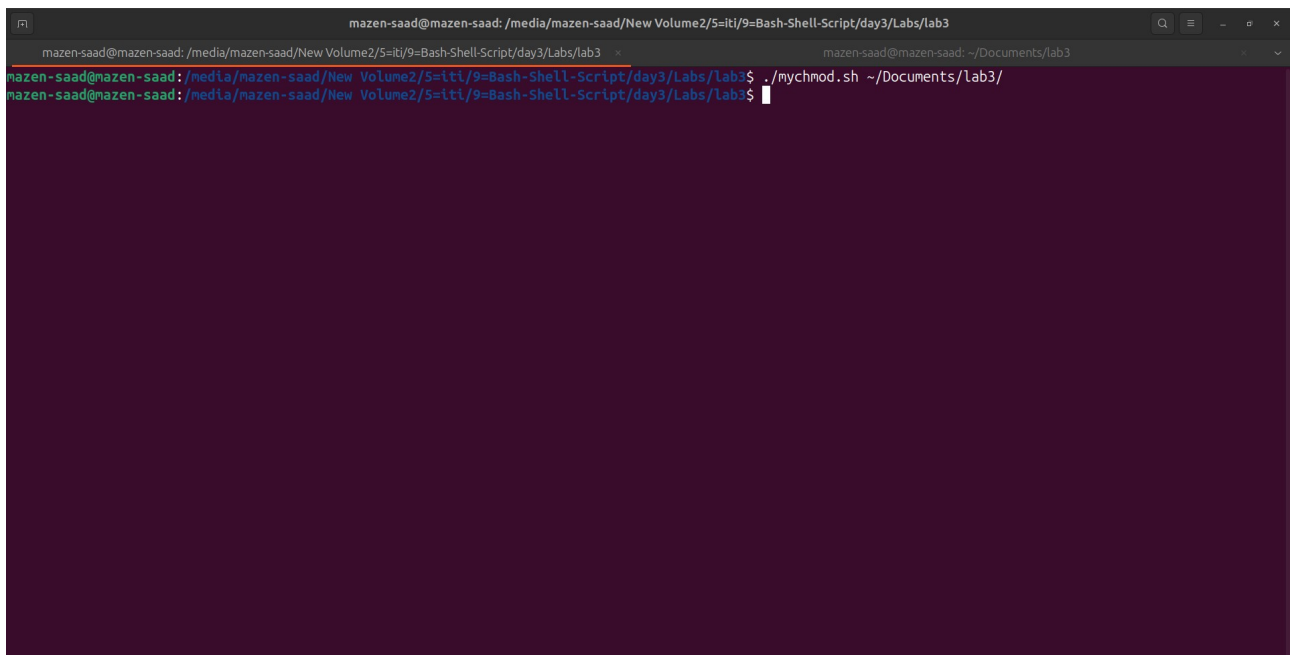
```
do
```

```
    chmod u+x "$item"
```

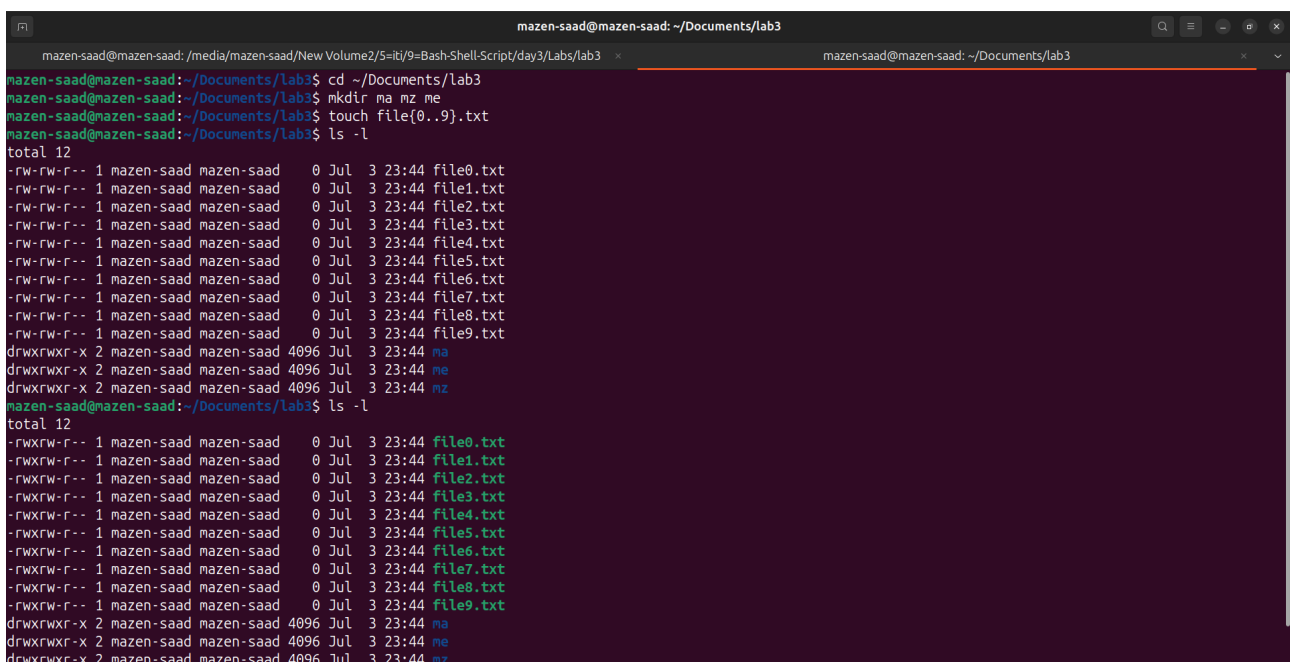
```
done
```

```
# Terminal =>
```

```
# ./mychmod.sh ~/Documents/lab3/
```



A terminal window with a dark purple background. The prompt is `mazen-saad@mazen-saad: ~/Documents/lab3`. The user has entered the command `./mychmod.sh ~/Documents/lab3/` and the cursor is at the end of the line.



A terminal window with a dark purple background. The prompt is `mazen-saad@mazen-saad: ~/Documents/lab3`. The user has entered the command `ls -l` and the output is displayed. The output shows the permissions for the files and directories created in the previous step. The files `file0.txt` through `file9.txt` have permissions `-rw-rw-r--`. The directories `ma`, `mz`, and `me` have permissions `drwxrwxr-x`.

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

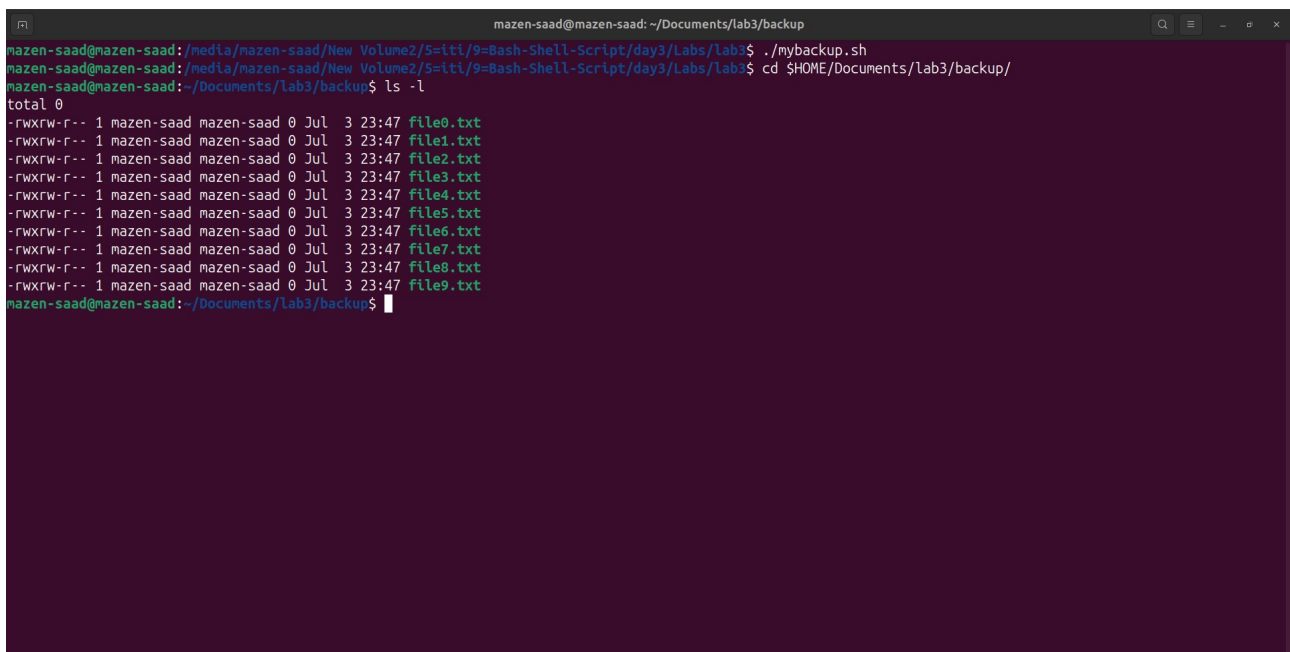
```
touch mybackup.sh
cd ~/Documents/lab3
mkdir ma mz me
touch file{0..9}.txt
```

```
#!/usr/bin/bash
shopt -s extglob
```

```
backup_dir="$HOME/Documents/lab3/backup/"
mkdir -p $backup_dir
```

```
for file in ~/Documents/lab3/*
do
    if [[ -f $file ]]
    then
        cp "$file" "$backup_dir"
    fi
done
```

```
# Terminal =>
# ./mybackup.sh
# cd $HOME/Documents/lab3/backup/
# ls -l
```



```
mazen-saad@mazen-saad: ~/Documents/lab3/backup
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mybackup.sh
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ cd $HOME/Documents/lab3/backup/
mazen-saad@mazen-saad: ~/Documents/lab3/backup$ ls -l
total 0
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file0.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file1.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file2.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file3.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file4.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file5.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file6.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file7.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file8.txt
-rwxrwxr-- 1 mazen-saad mazen-saad 0 Jul 3 23:47 file9.txt
mazen-saad@mazen-saad: ~/Documents/lab3/backup$
```

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.

```
sudo apt-get install mailutils  
=> Local only
```

```
touch mymail.sh mtemplate.txt
```

```
#!/usr/bin/bash  
shopt -s extglob
```

```
mtemplate=$1  
if [ ! -f $mtemplate ]  
then  
    echo "mtemplate file not found!"  
    exit 1  
fi
```

```
email_body=$(cat $mtemplate)
```

```
user_list=$(cut -d: -f1 /etc/passwd)
```

```
for user in $user_list  
do  
    mail -s "Subject of the email" "$user" <<< "$email_body"  
done
```

```
echo "Email has been sent to all users."
```

```
# Terminal =>  
# ./mymail.sh mtemplate.txt  
# cd /var/mail  
# ls -l
```

```
mazen-saad@mazen-saad: /var/mail
mazen-saad@mazen-saad:/media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$ ./mymail.sh mtemplate.txt
Email has been sent to all users.
mazen-saad@mazen-saad:/media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$ cd /var/mail
mazen-saad@mazen-saad:/var/mail$ ls -l
total 208
-rw-r----- 1 _apt mail 1924 Jul 3 23:50 _apt
-rw-r----- 1 avahi mail 1936 Jul 3 23:50 avahi
-rw-r----- 1 backup mail 1948 Jul 3 23:50 backup
-rw-r----- 1 bin mail 1912 Jul 3 23:50 bin
-rw-r----- 1 colord mail 1948 Jul 3 23:50 colord
-rw-r----- 1 cups-browsed mail 2020 Jul 3 23:50 cups-browsed
-rw-r----- 1 cups-pk-helper mail 2044 Jul 3 23:50 cups-pk-helper
-rw-r----- 1 daemon mail 1948 Jul 3 23:50 daemon
-rw-r----- 1 dhcpcd mail 1948 Jul 3 23:50 dhcpcd
-rw-r----- 1 dnsmasq mail 1960 Jul 3 23:50 dnsmasq
-rw-r----- 1 fwupd-refresh mail 2032 Jul 3 23:50 fwupd-refresh
-rw-r----- 1 games mail 1936 Jul 3 23:50 games
-rw-r----- 1 gdm mail 1912 Jul 3 23:50 gdm
-rw-r----- 1 geoclue mail 1960 Jul 3 23:50 geoclue
-rw-r----- 1 gnome-initial-setup mail 2104 Jul 3 23:50 gnome-initial-setup
-rw-r----- 1 gnome-remote-desktop mail 2116 Jul 3 23:50 gnome-remote-desktop
-rw-r----- 1 hplip mail 1936 Jul 3 23:50 hplip
-rw-r----- 1 irc mail 1912 Jul 3 23:50 irc
-rw-r----- 1 kernoops mail 1972 Jul 3 23:50 kernoops
-rw-r----- 1 list mail 1924 Jul 3 23:50 list
-rw-r----- 1 lp mail 1900 Jul 3 23:50 lp
-rw-r----- 1 mail mail 1924 Jul 3 23:50 mail
-rw-r----- 1 man mail 1912 Jul 3 23:50 man
-rw-r----- 1 mazen-saad mail 1996 Jul 3 23:50 mazen-saad
-rw-r----- 1 messagebus mail 1996 Jul 3 23:50 messagebus
-rw-r----- 1 mysql mail 1936 Jul 3 23:50 mysql
-rw-r----- 1 news mail 1924 Jul 3 23:50 news
-rw-r----- 1 nm-openvpn mail 1996 Jul 3 23:50 nm-openvpn
-rw-r----- 1 nobody mail 1948 Jul 3 23:50 nobody
```


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

```
touch chkmail.sh
```

```
#!/usr/bin/bash
shopt -s extglob

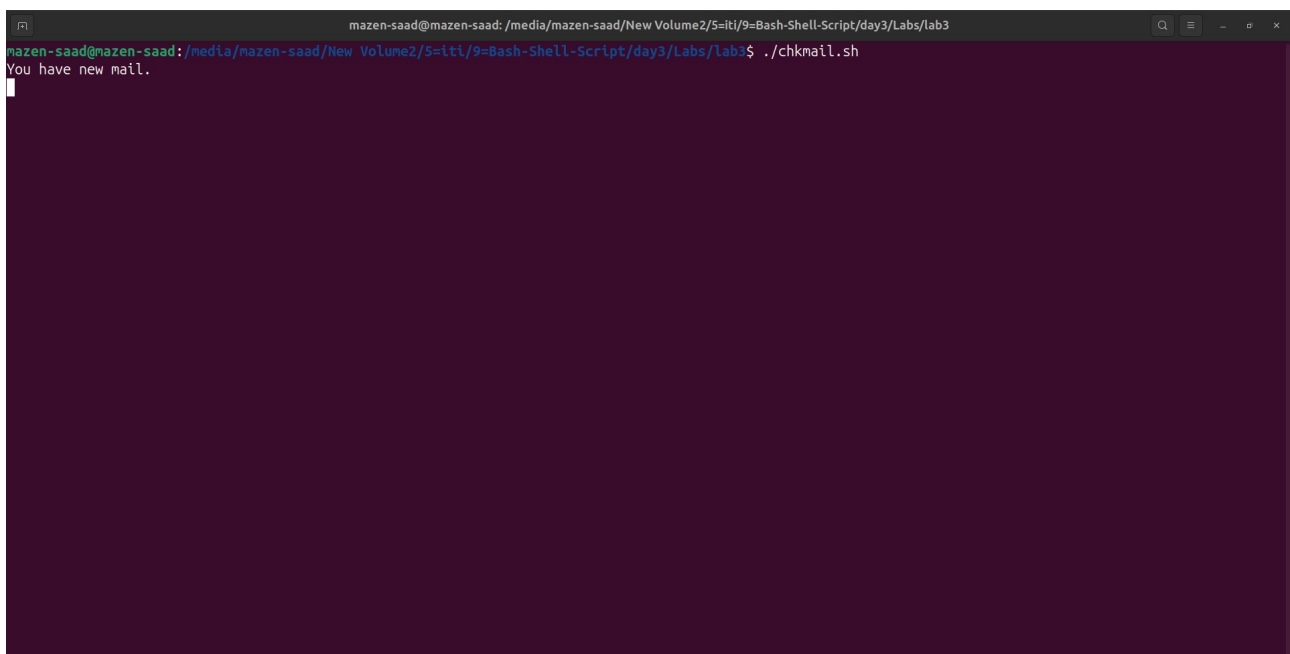
username=$(whoami)
mail_file="/var/mail/$username"

while true
do
    if [ -s "$mail_file" ]
    then
        echo "You have new mail."
    else
        echo "No new mail."
    fi
    sleep 10
done

# Terminal =>
# ./chkmail.sh
```

Bonus:

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

A terminal window with a dark background and light green text. The title bar shows the path "/media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3". The terminal content shows the command "mazen-saad@mazen-saad:/media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3\$./chkmail.sh" being executed, followed by the output "You have new mail." and a blank line.

```
mazen-saad@mazen-saad:/media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./chkmail.sh
You have new mail.
```

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
```

touch output.sh

```
#!/usr/bin/bash
shopt -s extglob
typeset -i n1
typeset -i n2
n1=1
n2=1

while test $n1 -eq $n2
do
    n2=$((n2+1))
    echo $n1

    if [ $n1 -gt $n2 ]
    then
        break
    else
        continue
    fi

    n1=$((n1+1))
    echo $n2
done

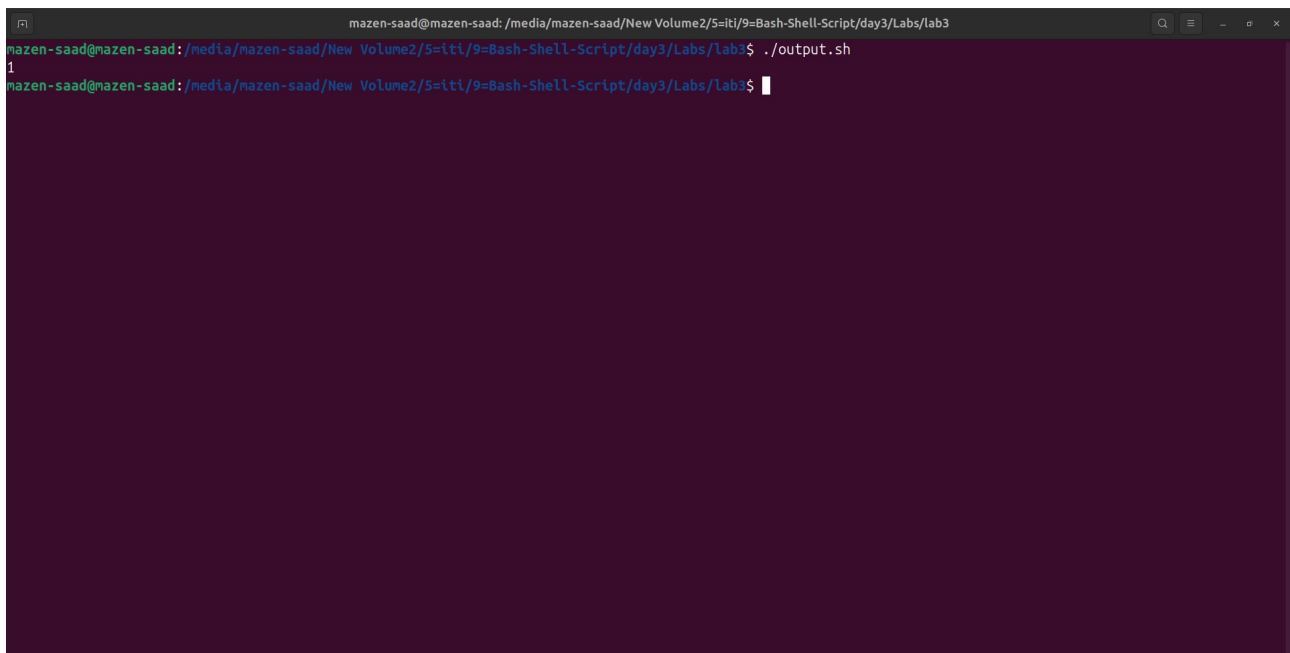
# 1 => Error
# ./output: line 1: typeset: `-'i': not a valid identifier
```

```
# ./output: line 2: typeset: `-'i': not a valid identifier
# ./output: line 5: test: -eq: binary operator expected
# Fix =>
# Replace -i with -i
```

```
# 2 => Error
# ./output: line 9: print: command not found
# Fix =>
# Replace print with echo
```

```
# result => 1
```

```
# Terminal =>
# ./output.sh
```

A terminal window with a dark background and light green text. The title bar shows the path: mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3. The terminal shows the command ./output.sh being executed, followed by the output 1. The prompt is mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3\$.

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$ ./output.sh
1
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$
```

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.

```
touch menu.sh
#!/usr/bin/bash
shopt -s extglob
```

```
PS3="Choose an option: "
```

```
select option in "ls" "ls -a" "Exit"
```

```
do
```

```
case $option in
```

```
"ls")
```

```
ls
```

```
;;
```

```
"ls -a")
```

```
ls -a
```

```
;;
```

```
"Exit")
```

```
break
```

```
;;
```

```
*)
```

```
echo "Invalid option."
```

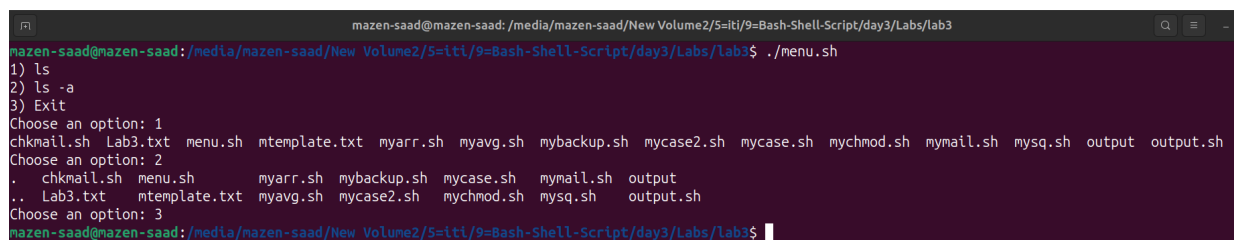
```
;;
```

```
esac
```

```
done
```

```
# Terminal =>
```

```
# ./menu.sh
```



```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$ ./menu.sh
1) ls
2) ls -a
3) Exit
Choose an option: 1
chkmail.sh Lab3.txt menu.sh mtemplate.txt myarr.sh myavg.sh mybackup.sh mycase2.sh mycase.sh mychmod.sh mymail.sh mysql.sh output output.sh
Choose an option: 2
. . . Lab3.txt mtemplate.txt myavg.sh mycase2.sh mychmod.sh mysql.sh output.sh
Choose an option: 3
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$
```

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.

```
touch myarr.sh
```

```
#!/usr/bin/bash  
shopt -s extglob
```

```
#!/usr/bin/bash  
shopt -s extglob
```

```
read -p "Please Enter Array Size: " size
```

```
for ((i=0;i<$size;i++))  
do
```

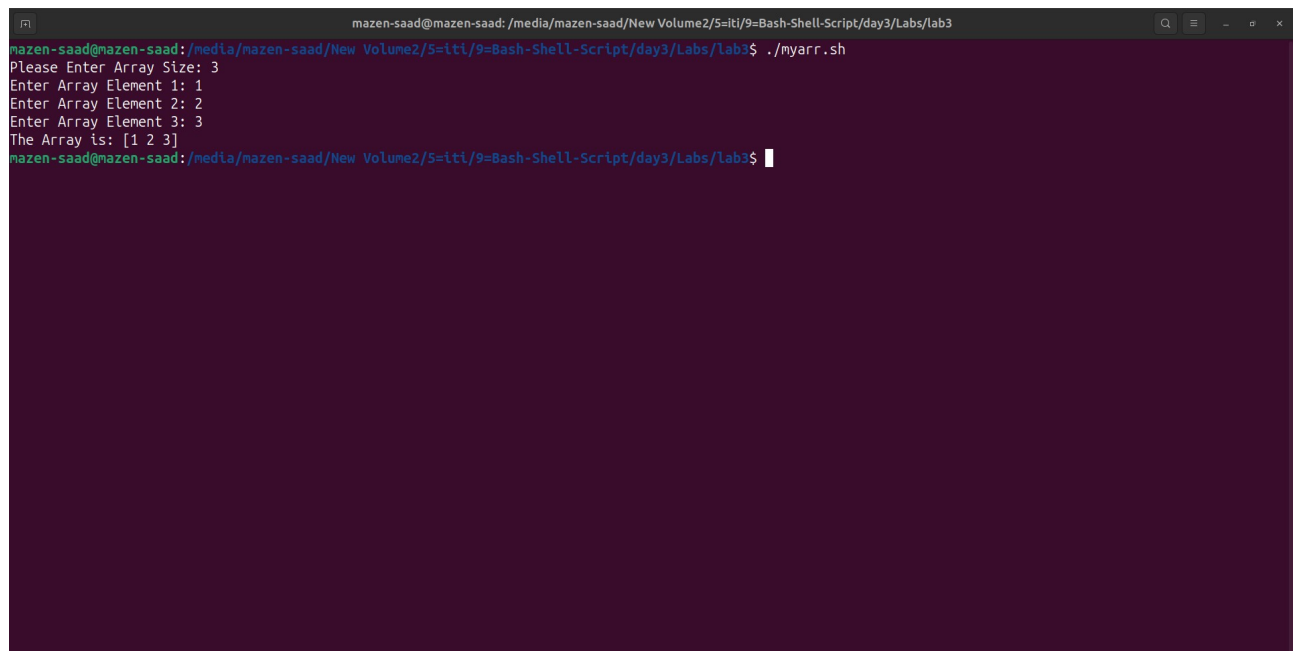
```
    read -p "Enter Array Element $((i+1)): " arr[$i]
```

```
done
```

```
echo "The Array is: [${arr[@]}]"
```

```
# Terminal =>
```

```
# ./myarr.sh
```

A terminal window with a dark purple background. The title bar shows the path "/media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3". The terminal content shows the execution of the script: the user enters '3' for the array size, then '1', '2', and '3' for the array elements. The script outputs "The Array is: [1 2 3]". The prompt "mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3\$" is visible at the bottom.

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3  
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./myarr.sh  
Please Enter Array Size: 3  
Enter Array Element 1: 1  
Enter Array Element 2: 2  
Enter Array Element 3: 3  
The Array is: [1 2 3]  
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$
```

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

```
touch myavg.sh
```

```
#!/usr/bin/bash  
shopt -s extglob
```

```
read -p "Please Enter Array Size: " size  
declare -i sum=0
```

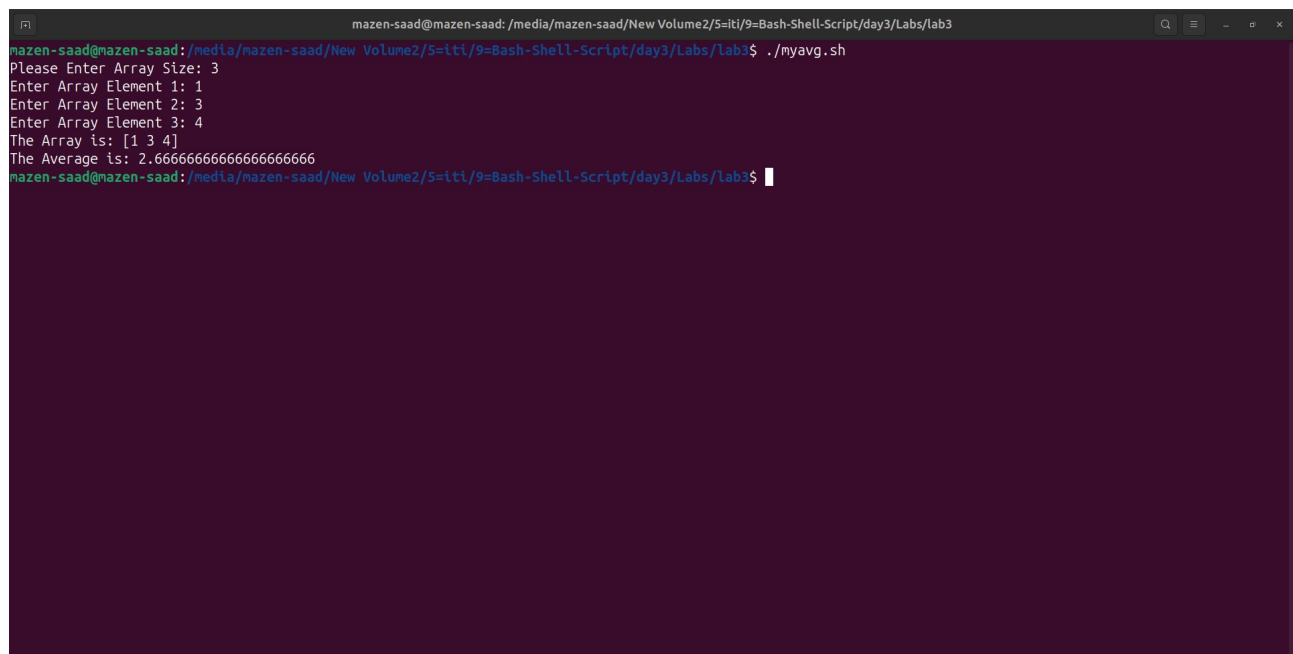
```
for ((i=0;i<$size;i++))  
do
```

```
    read -p "Enter Array Element $((i+1)): " arr[$i]  
    sum+=${arr[$i]}
```

```
done
```

```
echo "The Array is: ${arr[@]}"  
avg=$(echo "$sum / $size" | bc -l)  
echo The Average is: $avg
```

```
# Terminal =>  
# ./myavg.sh
```

A terminal window with a dark purple background. The title bar shows the path: mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3. The terminal output shows the script being executed: mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3\$./myavg.sh. The script prompts for the array size (3), then for three array elements (1, 3, 4). It then displays the array contents [1 3 4] and the calculated average 2.6666666666666666. The prompt returns to mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3\$.

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$ ./myavg.sh  
Please Enter Array Size: 3  
Enter Array Element 1: 1  
Enter Array Element 2: 3  
Enter Array Element 3: 4  
The Array is: [1 3 4]  
The Average is: 2.6666666666666666  
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=itl/9=Bash-Shell-Script/day3/Labs/lab3$
```

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

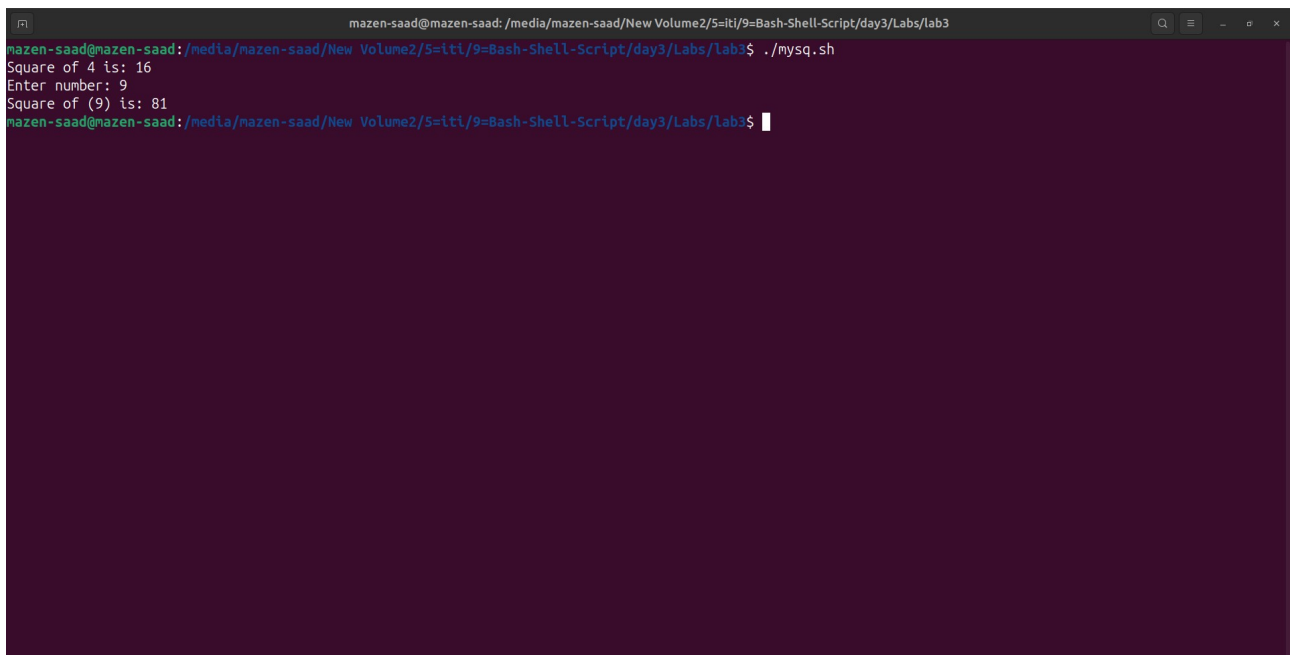
`touch mysq.sh`

```
#!/usr/bin/bash
shopt -s extglob
```

```
# =>1
function mysq(){
    echo $(( $1 * $1 ))
}
echo "Square of 4 is: $(mysq 4)"
```

```
# =>2
read -p "Enter number: " number
function mysq(){
    echo $(( $1 * $1 ))
}
echo "Square of ($number) is: $(mysq $number)"
```

```
# Terminal =>
# ./mysq.sh
```

A terminal window with a dark background and light green text. The window title is "mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3". The prompt is "mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3\$". The user has entered the command ". /mysq.sh". The output shows "Square of 4 is: 16", followed by a prompt "Enter number: " where the user has entered "9". The final output is "Square of (9) is: 81". The prompt is now "mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3\$".

```
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$ ./mysq.sh
Square of 4 is: 16
Enter number: 9
Square of (9) is: 81
mazen-saad@mazen-saad: /media/mazen-saad/New Volume2/5=iti/9=Bash-Shell-Script/day3/Labs/lab3$
```