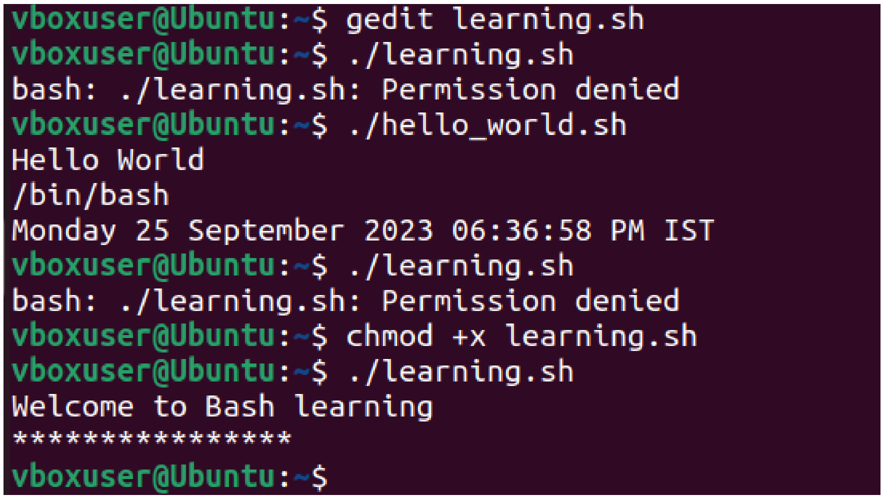
MODULE 3

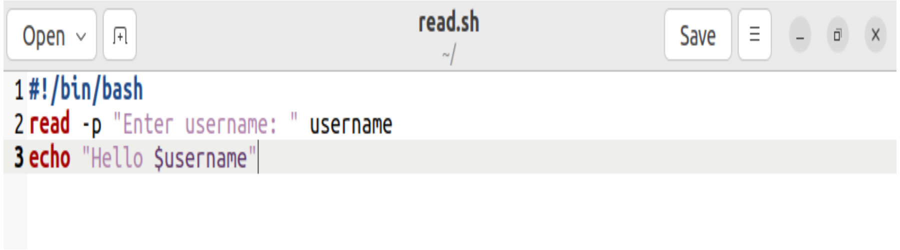
ASSIGNMENT 1

1) **Write a simple Bash shell script to display the message "Welcome to Bash learning" and "***************" on separate lines.**



```
vboxuser@Ubuntu:~$ gedit learning.sh
vboxuser@Ubuntu:~$ ./learning.sh
bash: ./learning.sh: Permission denied
vboxuser@Ubuntu:~$ ./hello_world.sh
Hello World
/bin/bash
Monday 25 September 2023 06:36:58 PM IST
vboxuser@Ubuntu:~$ ./learning.sh
bash: ./learning.sh: Permission denied
vboxuser@Ubuntu:~$ chmod +x learning.sh
vboxuser@Ubuntu:~$ ./learning.sh
Welcome to Bash learning
****************
vboxuser@Ubuntu:~$
```

read.sh



```
#!/bin/bash
read -p "Enter username: " username
echo "Hello $username"
```

ASSIGNMENT 2

1) Write a simple Bash program to get the following system variables:

a. pwd

b. logname

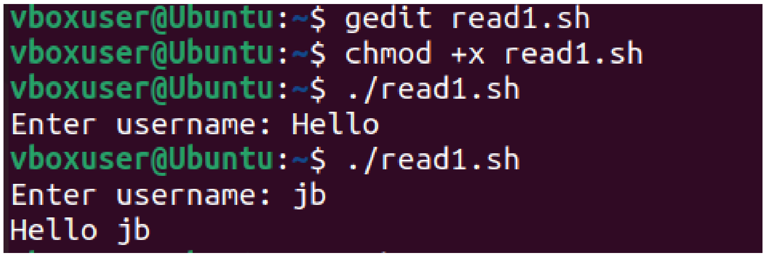
2) Write a simple Bash program:

a. To ask username from user

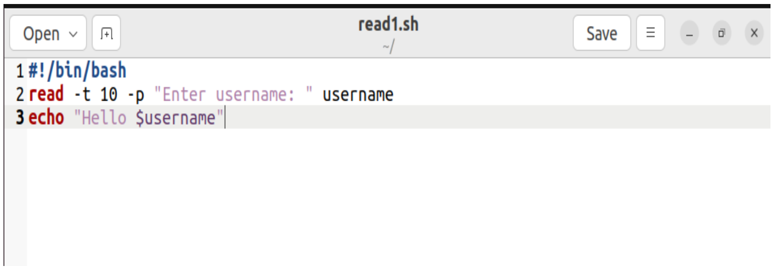
b. Exit the program, if user does not enter anything within 10 seconds

Hint: read -t 10 -p

2.

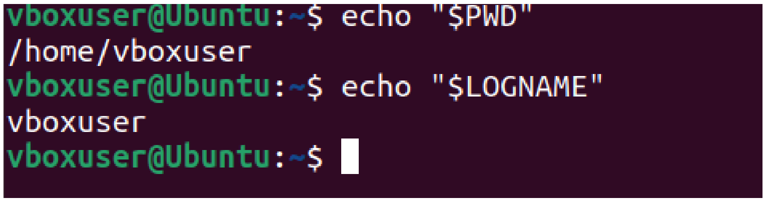


```
vboxuser@Ubuntu:~$ gedit read1.sh
vboxuser@Ubuntu:~$ chmod +x read1.sh
vboxuser@Ubuntu:~$ ./read1.sh
Enter username: Hello
vboxuser@Ubuntu:~$ ./read1.sh
Enter username: jb
Hello jb
```



```bash
#!/bin/bash
read -t 10 -p "Enter username: " username
echo "Hello $username"
```

1.

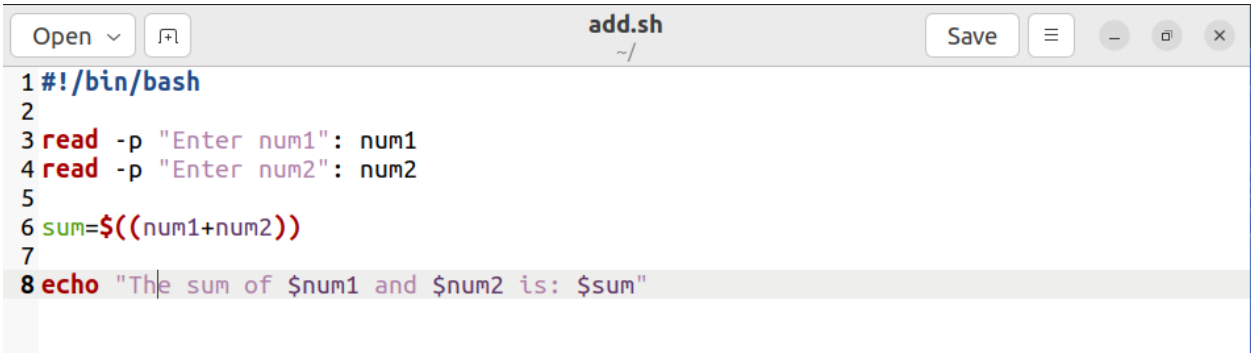


```
vboxuser@Ubuntu:~$ echo "$PWD"
/home/vboxuser
vboxuser@Ubuntu:~$ echo "$LOGNAME"
vboxuser
vboxuser@Ubuntu:~$ 
```

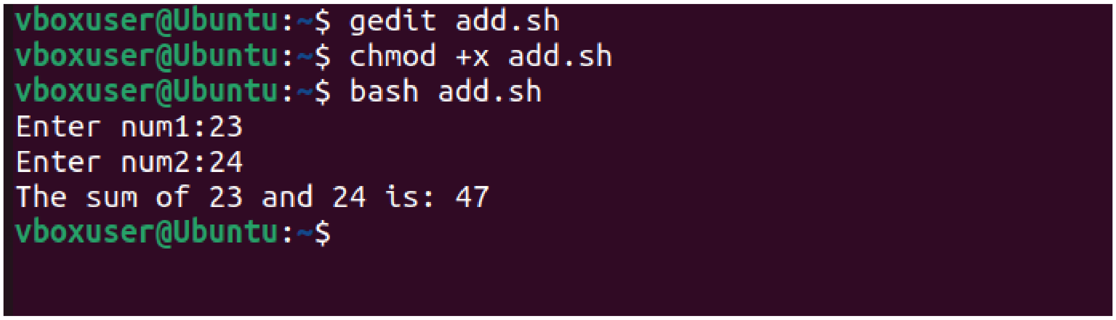
ASSIGNMENT 3

Command Line arguments and Quoting

1) Write a bash program for addition using command line arguments.



```
#!/bin/bash

read -p "Enter num1": num1
read -p "Enter num2": num2

sum=$((num1+num2))

echo "The sum of $num1 and $num2 is: $sum"
```



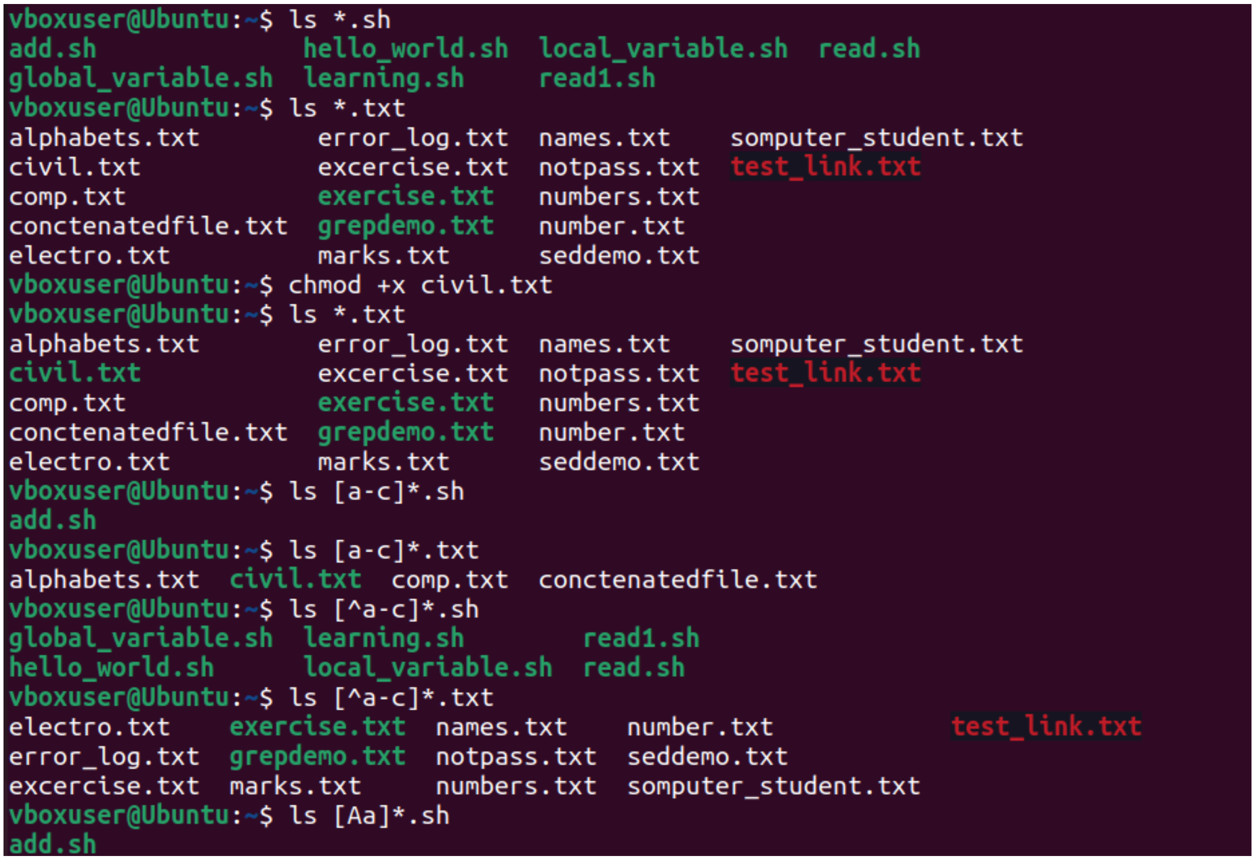
```
vboxuser@Ubuntu:~$ gedit add.sh
vboxuser@Ubuntu:~$ chmod +x add.sh
vboxuser@Ubuntu:~$ bash add.sh
Enter num1:23
Enter num2:24
The sum of 23 and 24 is: 47
vboxuser@Ubuntu:~$
```

ASSIGNMENT 4

Globbing and Export statement

1) Write a Bash script to do all operations discussed under Globbing

Concepts under globbing:



```
vboxuser@Ubuntu:~$ ls *.sh
add.sh              hello_world.sh  local_variable.sh  read.sh
global_variable.sh  learning.sh     read1.sh
vboxuser@Ubuntu:~$ ls *.txt
alphabets.txt        error_log.txt  names.txt    somputer_student.txt
civil.txt            excercise.txt  notpass.txt  test_link.txt
comp.txt             exercise.txt   numbers.txt
conctenatedfile.txt  grepdemo.txt   number.txt
electro.txt          marks.txt      seddemo.txt
vboxuser@Ubuntu:~$ chmod +x civil.txt
vboxuser@Ubuntu:~$ ls *.txt
alphabets.txt        error_log.txt  names.txt    somputer_student.txt
civil.txt            excercise.txt  notpass.txt  test_link.txt
comp.txt             exercise.txt   numbers.txt
conctenatedfile.txt  grepdemo.txt   number.txt
electro.txt          marks.txt      seddemo.txt
vboxuser@Ubuntu:~$ ls [a-c]*.sh
add.sh
vboxuser@Ubuntu:~$ ls [a-c]*.txt
alphabets.txt  civil.txt  comp.txt  conctenatedfile.txt
vboxuser@Ubuntu:~$ ls [^a-c]*.sh
global_variable.sh  learning.sh        read1.sh
hello_world.sh      local_variable.sh  read.sh
vboxuser@Ubuntu:~$ ls [^a-c]*.txt
electro.txt    exercise.txt  names.txt    number.txt            test_link.txt
error_log.txt  grepdemo.txt  notpass.txt  seddemo.txt
excercise.txt  marks.txt     numbers.txt  somputer_student.txt
vboxuser@Ubuntu:~$ ls [Aa]*.sh
add.sh
```

ASSIGNMENT 5

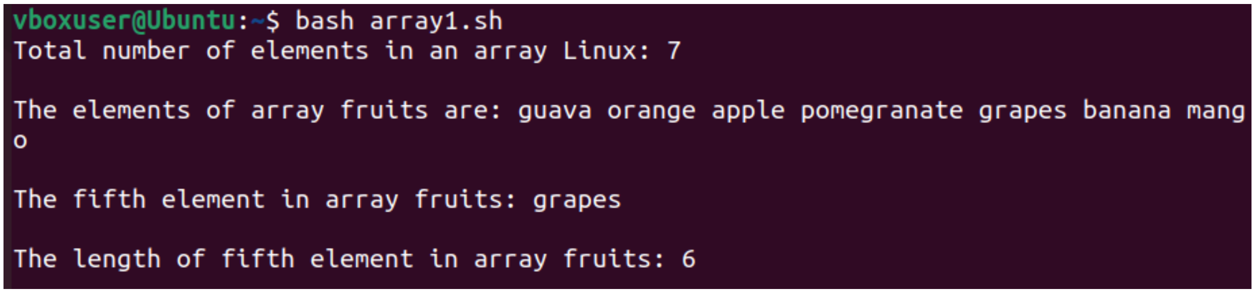
Array Operations in BASH

1) Declare an Array names of length 7 and find

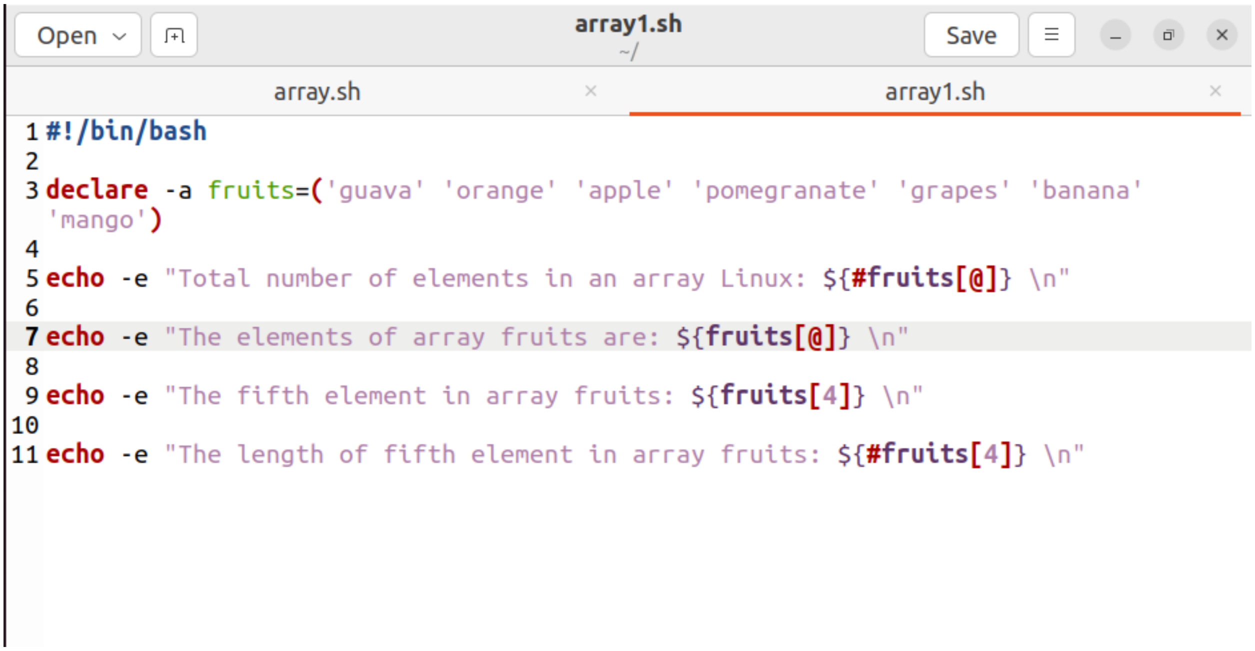
a. The total number of elements

b. Print all the elements

c. Print the 5th element



```
vboxuser@Ubuntu:~$ bash array1.sh
Total number of elements in an array Linux: 7

The elements of array fruits are: guava orange apple pomegranate grapes banana mang
o

The fifth element in array fruits: grapes

The length of fifth element in array fruits: 6
```



```bash
#!/bin/bash

declare -a fruits=('guava' 'orange' 'apple' 'pomegranate' 'grapes' 'banana' 'mango')

echo -e "Total number of elements in an array Linux: ${#fruits[@]} \n"

echo -e "The elements of array fruits are: ${fruits[@]} \n"

echo -e "The fifth element in array fruits: ${fruits[4]} \n"

echo -e "The length of fifth element in array fruits: ${#fruits[4]} \n"
```

ASSIGNMENT 6

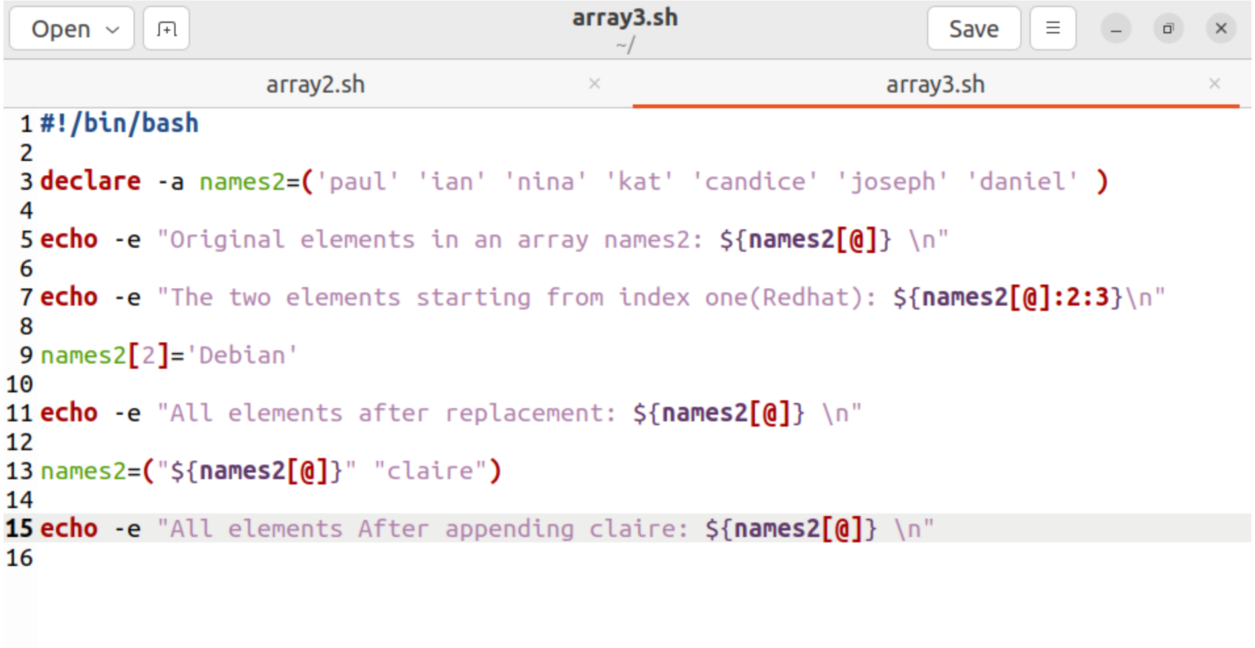
More on Arrays

1) Declare an Array names2 of length 7 and perform following operations-

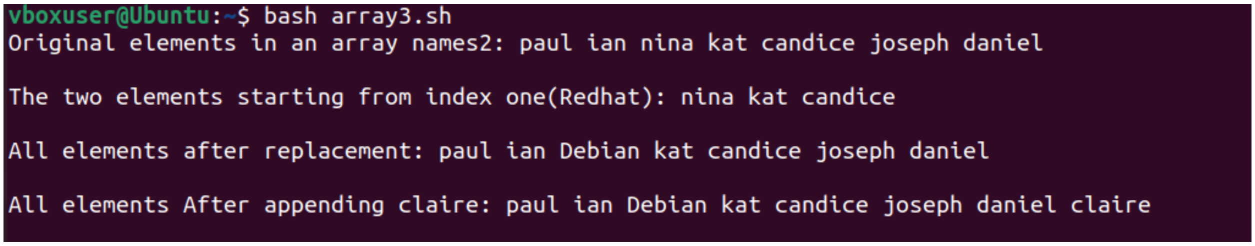
a. Extract three elements starting from index two.

b. Replace third element with 'Debian' and display.

c. Append any new name at the end of Array.



```
#!/bin/bash

declare -a names2=('paul' 'ian' 'nina' 'kat' 'candice' 'joseph' 'daniel' )

echo -e "Original elements in an array names2: ${names2[@]} \n"

echo -e "The two elements starting from index one(Redhat): ${names2[@]:2:3}\n"

names2[2]='Debian'

echo -e "All elements after replacement: ${names2[@]} \n"

names2=("${names2[@]}" "claire")

echo -e "All elements After appending claire: ${names2[@]} \n"

```



```
vboxuser@Ubuntu:~$ bash array3.sh
Original elements in an array names2: paul ian nina kat candice joseph daniel

The two elements starting from index one(Redhat): nina kat candice

All elements after replacement: paul ian Debian kat candice joseph daniel

All elements After appending claire: paul ian Debian kat candice joseph daniel claire
```

ASSIGNMENT 7

Conditional execution

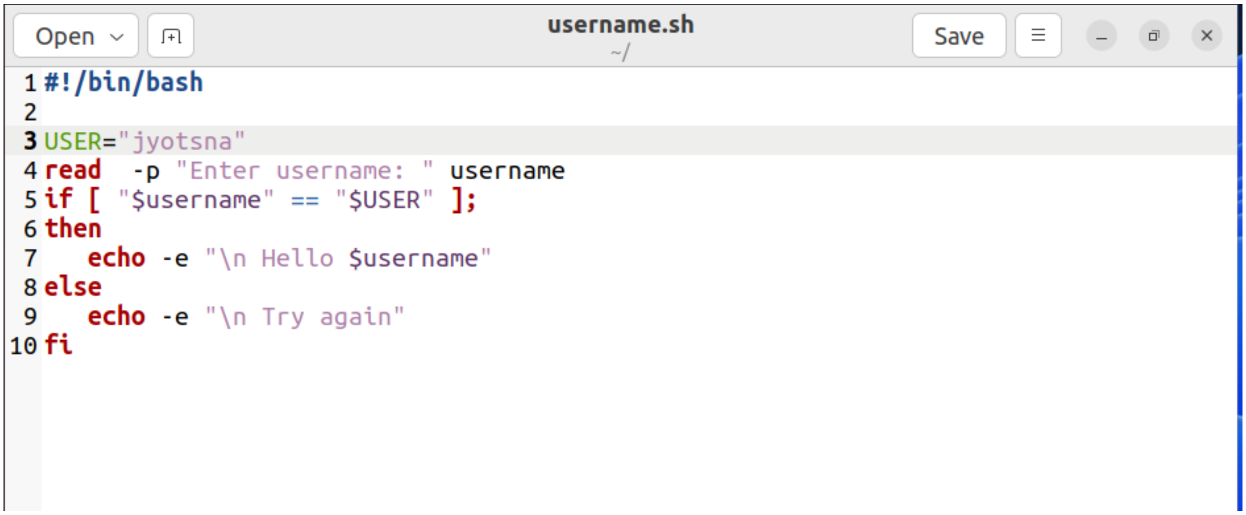
1) Write a script which will take your name as an input.

2) It should check this name with your system's username.

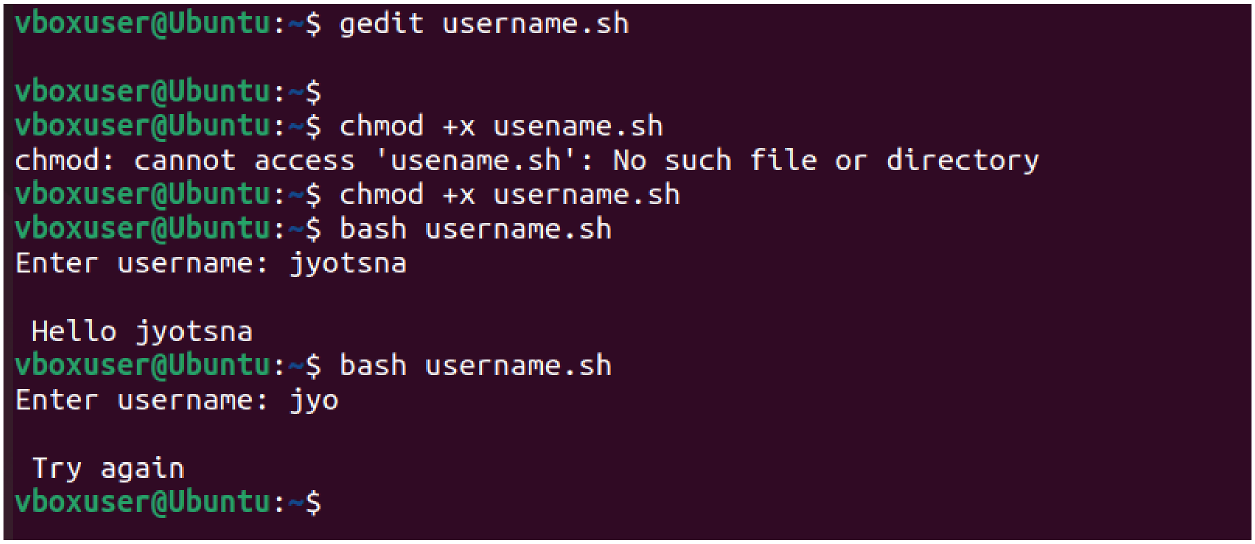
3) If the username matches, it should greet you by displaying "Hello".

4) Else, it should display "Try again"

HINT: Your system's username is stored in a variable $USER



```bash
#!/bin/bash

USER="jyotsna"
read  -p "Enter username:  " username
if [ "$username" == "$USER" ];
then
   echo -e "\n Hello $username"
else
   echo -e "\n Try again"
fi
```



```
vboxuser@Ubuntu:~$ gedit username.sh

vboxuser@Ubuntu:~$
vboxuser@Ubuntu:~$ chmod +x usename.sh
chmod: cannot access 'usename.sh': No such file or directory
vboxuser@Ubuntu:~$ chmod +x username.sh
vboxuser@Ubuntu:~$ bash username.sh
Enter username: jyotsna

 Hello jyotsna
vboxuser@Ubuntu:~$ bash username.sh
Enter username: jyo

 Try again
vboxuser@Ubuntu:~$
```

ASSIGNMENT 8

Nested and multilevel if elseif statements

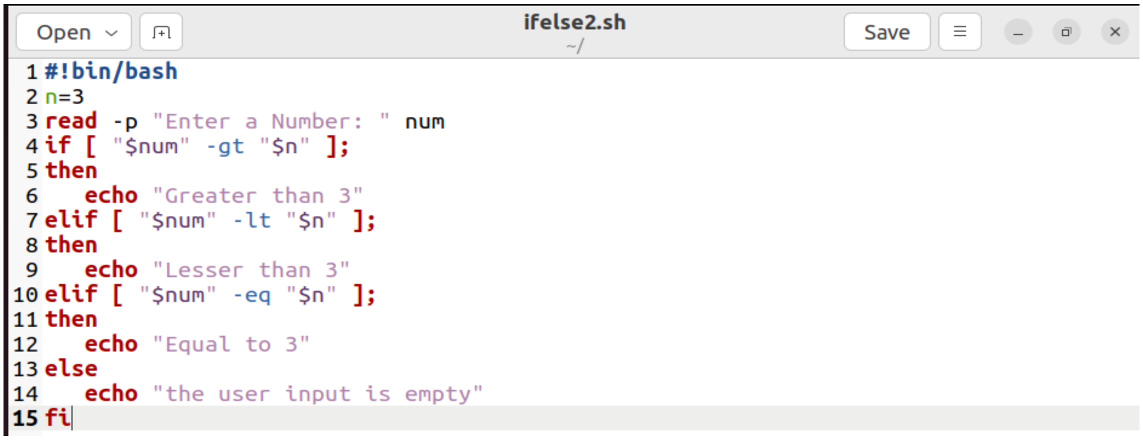
1) Write a program to output different messages when number is:

a. Greater than 3

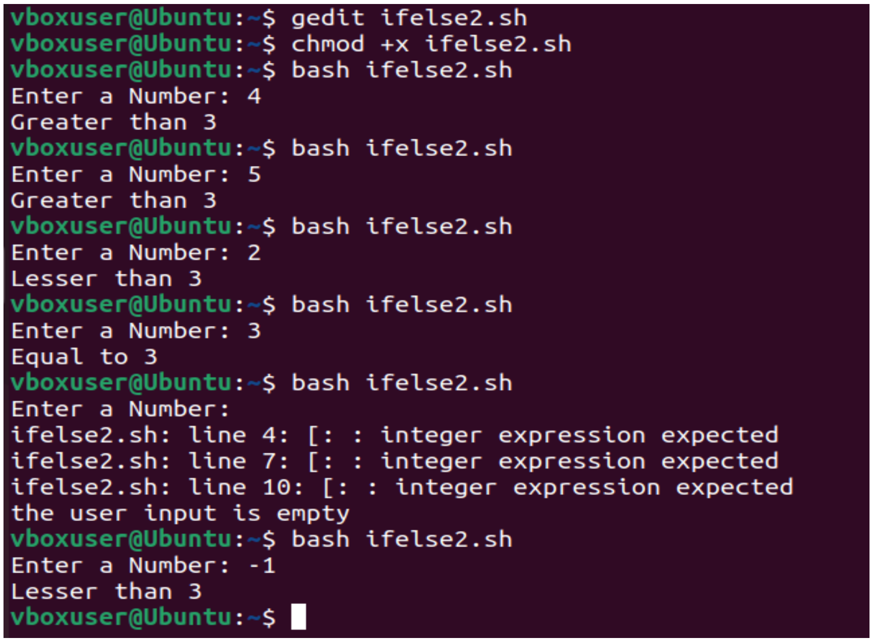
b. Lesser than 3

c. Or equal to 3

d. Or when the user input is empty



```
 1 #!bin/bash
 2 n=3
 3 read -p "Enter a Number: " num
 4 if [ "$num" -gt "$n" ];
 5 then
 6     echo "Greater than 3"
 7 elif [ "$num" -lt "$n" ];
 8 then
 9     echo "Lesser than 3"
10 elif [ "$num" -eq "$n" ];
11 then
12     echo "Equal to 3"
13 else
14     echo "the user input is empty"
15 fi
```



```
vboxuser@Ubuntu:~$ gedit ifelse2.sh
vboxuser@Ubuntu:~$ chmod +x ifelse2.sh
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number: 4
Greater than 3
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number: 5
Greater than 3
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number: 2
Lesser than 3
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number: 3
Equal to 3
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number:
ifelse2.sh: line 4: [: : integer expression expected
ifelse2.sh: line 7: [: : integer expression expected
ifelse2.sh: line 10: [: : integer expression expected
the user input is empty
vboxuser@Ubuntu:~$ bash ifelse2.sh
Enter a Number: -1
Lesser than 3
vboxuser@Ubuntu:~$ 
```