Experiment- 7

Bash Script / Shell Programming

Creating a Script

Step 1. Create file **filename.sh**
Step 2. Write content in filename.sh with starting line **#! /bin/bash**
Step 3. Change mode of file to executable by command **chmod u+x filename.sh**
Step 4. Execute the file by command **./filename.sh**

```
tryhackme@linux1:~$ rm test1.sh
tryhackme@linux1:~$ ls -l
total 84
-rw-rw-r-- 1 tryhackme tryhackme 65522 May 10  2021 access.log
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder1
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder2
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder3
drwxr-xr-x 2 tryhackme tryhackme  4096 May 10  2021 folder4
-rwxrwxrwx 1 tryhackme tryhackme    23 Oct  9 08:48 test.sh
tryhackme@linux1:~$ cat> test1.sh
#!/bin/bash
echo "hello"
^C
tryhackme@linux1:~$ chmod 777 test1.sh
tryhackme@linux1:~$ ./test1.sh
hello
tryhackme@linux1:~$ read NAME
```

2. **Reading User Inputs**

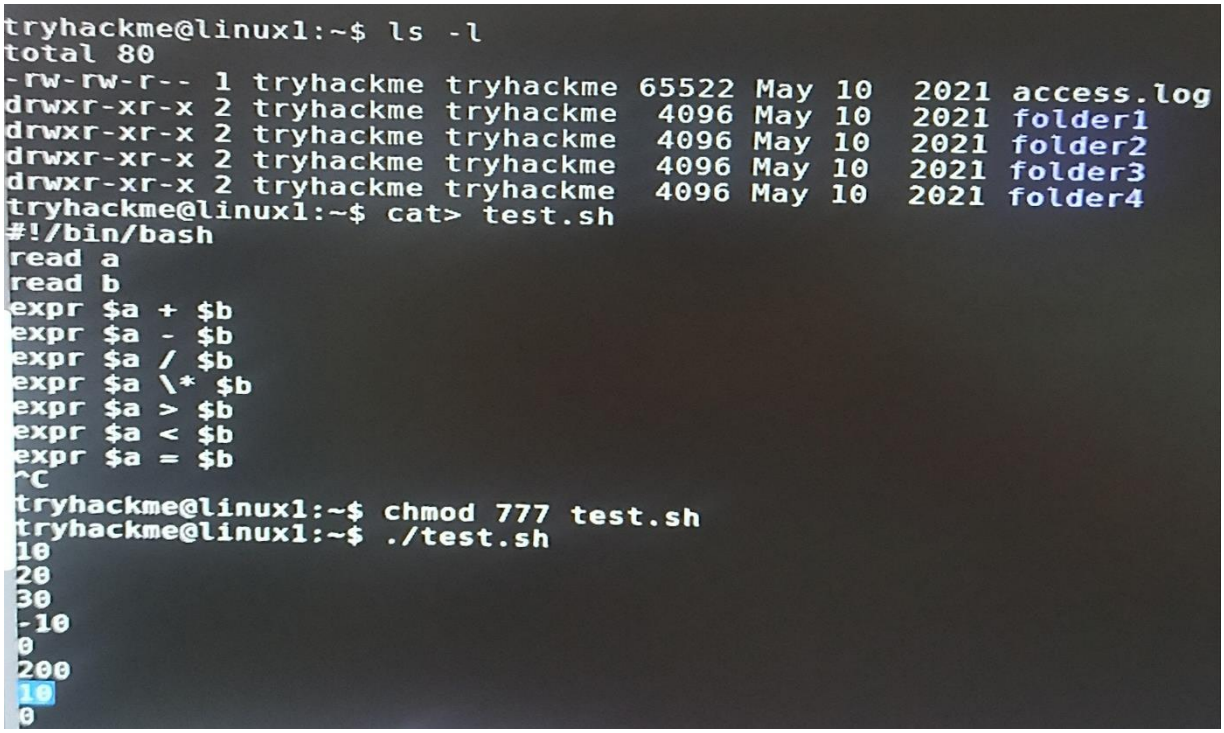
```
tryhackme@linux1:~$ cat> test.sh
#!/bin/bash
echo "hello "
read NAME
^C
tryhackme@linux1:~$ chmod 777 test.sh
tryhackme@linux1:~$ ./test.sh
hello
jaydeep
tryhackme@linux1:~$ cat test.sh
#!/bin/bash
echo "hello "
read NAME
tryhackme@linux1:~$ ./test.sh
hello
Jaydeep
tryhackme@linux1:~$ cat test.sh
#!/bin/bash
echo "hello "
read NAME
tryhackme@linux1:~$ ./test.sh
hello
10
tryhackme@linux1:~$
```

Steps: Take Input from user

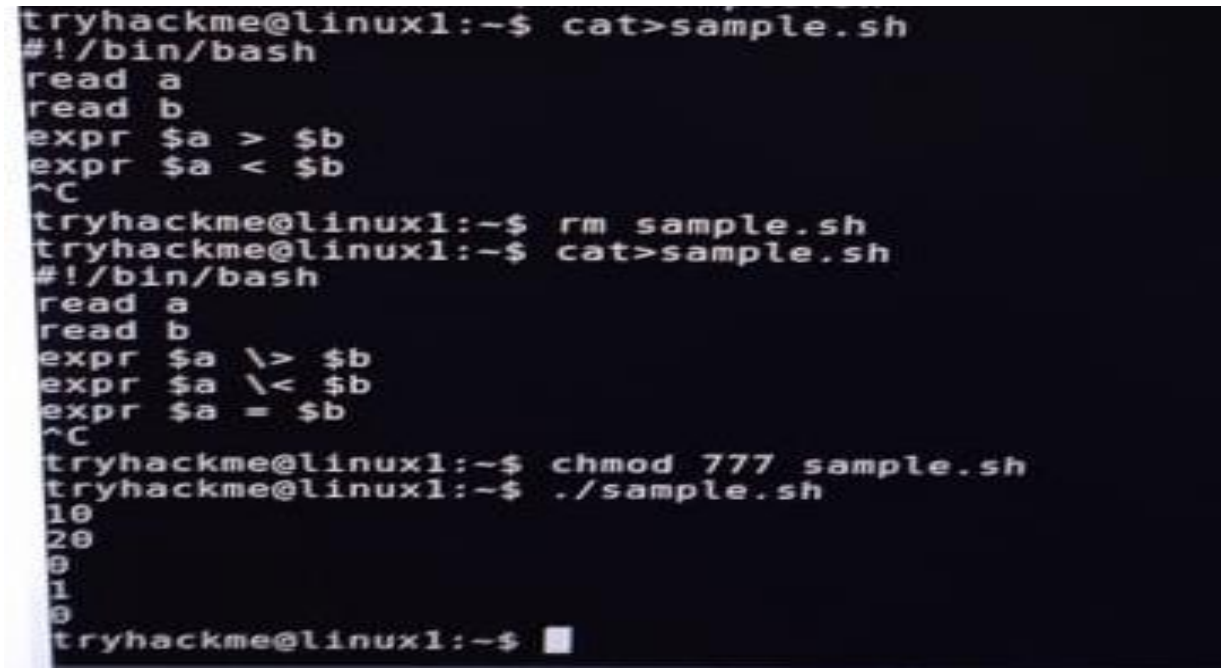
read input name

If we want to execute the use **$input name**

3. **expr:** It is used to write expressions.



4. **Logical expressions**



Experiment- 8

Conditional Statements and Looping in Shell Script

1. **Conditional Statements:** There are 5 conditional statements which can be used in bash programming
    2. if statement
    3. if-else statement
    4. if..elif..else..fi statement (Else If ladder)
    5. if..then..else..if..then..fi..fi..(Nested if)
    6. switch statement

if statement

This block will process if specified condition is true.

if-else statement

If specified condition is not true in if part then else part will be executed.

if..elif..else..fi statement (Else If ladder)

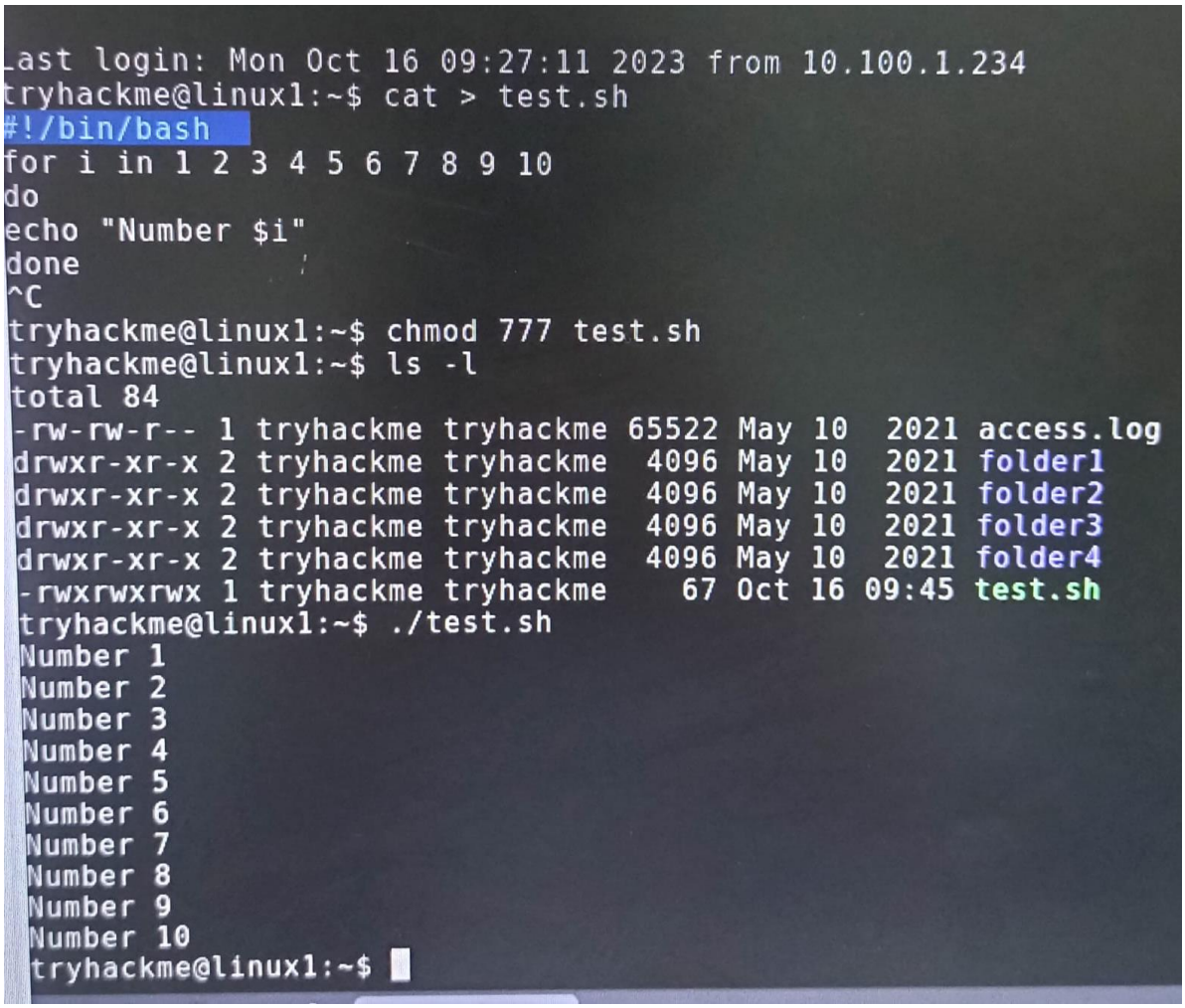
To use multiple conditions in one if-else block, then elif keyword is used in shell. If expression1 is true then it executes statement 1 and 2, and this process continues. If none of the condition is true then it processes else part.

```
The list of available updates is more t
To check for new updates run: sudo apt

tryhackme@linux1:~$ cat > num.sh
#!/bin/bash
echo "Enter a number "
read num
if [ $((num % 2)) -eq 0 ];
then
echo "EVEN"
else
echo "ODD"
fi
tryhackme@linux1:~$ chmod +x num.sh
tryhackme@linux1:~$ ./num.sh
Enter a number
7
ODD
tryhackme@linux1:~$ ./num.sh
Enter a number
2
EVEN
tryhackme@linux1:~$
```

2. **Looping Statements in Shell Scripting:** There are total 3 looping statements that can be used in bash programming

1. **while statement**:
   Here the command is evaluated and based on the resulting loop will execute, if the command is raised to false then the loop will be terminated.
2. **for statement**:
   The for loop operates on lists of items. It repeats a set of commands for every item in a list.
   Here var is the name of a variable and word1 to word N are sequences of characters separated by spaces (words). Each time the for loop executes, the value of the variable var is set to the next word in the list of words, word1 to word N.
3. **until statement:**
   The until loop is executed as many times as the condition/command evaluates too false. The loop terminates when the condition/command becomes true.



```
Last login: Sat Oct 28 05:45:33 2023 from 10.100.2.80
tryhackme@linux1:~$ cat > table.sh
#!/bin/bash
echo "Enter a number: "
read num
counter=1
while [
tryhackme@linux1:~$ rm table.sh
tryhackme@linux1:~$ cat > table.sh
#!/bin/bash
echo "Enter a number "
read num
counter=1
while [ $counter -le 10 ];
do
product=$((num * counter))
echo "$num x $counter = $product"
counter=$((counter + 1))
done
tryhackme@linux1:~$ chmod +x table.sh
tryhackme@linux1:~$ ./table.sh
Enter a number
5
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
tryhackme@linux1:~$
```