Experiment 06

Aim: To study the usage of For loop in shell.

- Using a for loop in shell scripting can be handy for iterating through lists of items or performing operations on files.

- In shell scripting, for loops typically follow this syntax:

```
for item in list
    do
        #commands to be executed for each item
    done
```

Program Case 1: Echo Basic Manage

```
vi forloop1.sh
#!/bin/bash
SERVERS="s1 s2 s3"
for S in $SERVERS; do
    echo "updating pkg on: $S"
done
~
~
chmod u+rwx forloop1.sh
./forloop1.sh
```

```
localhost:~# vi forloop1.sh
#!/bin/bash
SERVERS="s1 s2 s3"
for S in $SERVERS; do
        echo "updating pkg on: $S"
done
~
~
~
localhost:~# chmod u+rwx forloop1.sh
localhost:~# ./forloop1.sh
updating pkg on: s1
updating pkg on: s2
updating pkg on: s3
```

Program Case 2: Iterating through range of Numbers

```
vi for2.sh
#!/bin/bash
for value in {1..5}
do
    echo "number: $value"
done
~
~
chmod u+rwx for2.sh
./for2.sh
```

```
localhost:~# vifor2.sh
#!/bin/bash
for value in {1..5}
do
    echo "number: $value"
done
~
~
localhost:~# chmod u+rwx for2.sh
localhost:~# ./for2.sh
number: 1
number: 2
number: 3
number: 4
number: 5
```

Program Case 3: Iterating through multiple files

```
vi forloop3.sh
#!/bin/bash
for file in /root/*
do
   chmod 755 "$file"
   echo "update permission for: $file"
done
~
chmod u+rwx forloop3.sh
./forloop3.sh
```

```
localhost:~# vi forloop3.sh
#!/bin/bash
for file in /root/*
do
    chmod 755 "$file"
    echo "update permission for: $file"
done
~
~
localhost:~# chmod u+rwx forloop3.sh
localhost:~# ./forloop3.sh
update permission for: /root/bench.py
update permission for: /root/forloop3.sh
update permission for: /root/hello.c
update permission for: /root/hello.js
update permission for: /root/readme.txt
```

Program Case 4: Creating an Infinite Loop

```
vi forloop4.sh
#!/bin/bash
for (( ; ; ))
do
   echo "This is infinite loop"
   echo "Use Ctrl+C to stop it"
done
~
~
Chmod u+rwx forloop4.sh
./forloop4.sh
```
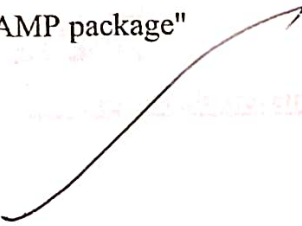
```
localhost:~# vi forloop4.sh
#!/bin/bash
for (( ; ; ))
do
    echo "This is infinite loop"
    echo "Use Ctrl+C to stop it"
done
~
~
localhost:~# chmod u+rwx forloop4.sh
localhost:~# ./forloop4.sh
This is infinite loop
Use Ctrl+C to stop it
This is infinite loop
Use Ctrl+C to stop it
This is infinite loop
Use Ctrl+C to stop it
This is infinite loop
^C
localhost:~#
```

Program Case 5: Implementing a Nested for loop

```
vi forloop5.sh
#!/bin/bash
for serverd in A B C; do
   for app in apache dp; do
      echo "$serverd can run $app LAMP package"
   done
done
~
~
Chmod 711 forloop5.sh
./forloop5.sh
```

```
localhost:~# vi forloop5.sh
#!/bin/bash
for serverd in A B C; do
    for app in apache dp; do
        echo "$serverd can run $app LAMP package"
    done
done
~
~
localhost:~# chmod 711 forloop5.sh
localhost:~# ./forloop5.sh
A can run apache LAMP package
A can run dp LAMP package
B can run apache LAMP package
B can run dp LAMP package
C can run apache LAMP package
C can run dp LAMP package
```

Program Case 6: Array utilization in for loop

```
vi forloop6.sh
#!/bin/bash
apps=("apache" "mysql" "php")
for app in "${apps[@]}"
do
    echo "The application name is $app"
done
~
~

chmod 711 forloop6.sh
./forloop6.sh
```

```
localhost:~# vi forloop6.sh
#!/bin/bash
apps=("apache" "mysql" "php")
for app in "${apps[@]}"
do
    echo "The application name is $app"
done
~
~
localhost:~# chmod 711 forloop6.sh
localhost:~# ./forloop6.sh
The application name is apache
The application name is mysql
The application name is php
```

Program Case 7: Using break statement in for loop

```
vi forloop7.sh
#!/bin/bash
for file in ~/.* ; do
    if [[ "$file" == "./data.txt" ]]
    then
        echo "$file is available"
        break
    fi
done
~
chmod 711 forloop7.sh
./forloop7.sh
```

```
localhost:~# vi forloop7.sh
#!/bin/bash
for file in ~/.* ; do
    if [[ "$file" == "./data.txt" ]]
    then
        echo "$file is available"
        break
    fi
done
~
~
localhost:~# chmod 711 forloop7.sh
localhost:~# ./forloop7.sh
```

Program Case 8: Use of command substitution

```
vi forloop8.sh
#!/bin/bash
for log in $(cat ~/testfile)
do
    echo "Log entry: $log"
done
~
~
chmod 711 forloop8.sh
./forloop8.sh
```

```
localhost:~# vi forloop8.sh
#!/bin/bash

for log in $(cat ~/testfile)
do
        echo "Log entry:..$log"
done
~
~
~
localhost:~# chmod 711 forloop8.sh
localhost:~# ./forloop8.sh
cat: can't open '/root/testfile': No such file or directory
```

3. WAP to create directories through vi text editor

```
localhost:~# vi directories.sh
#!/bin/bash
mkdir -p {Maths, English}/{Notes, Examresults}
~
~
~

localhost:~/Ritika# chmod +x directories.sh
localhost:~/Ritika# ./directories.sh
localhost:~/Ritika# ls -R
.:
English}            Examresults)    directories.sh  {Maths,

./English}:
{Notes,

./English}/{Notes,:

./Examresults}:

./{Maths,:
```

4. WAP to read a file into a variable

```
localhost:~/Ritika# cat > mysamplefile.txt
This is exp 8 of OS Lab.
localhost:~/Ritika# vi readfile.sh

#!/bin/bash
myvalue=$(cat mysamplefile.txt)
echo "$myvalue"

localhost:~/Ritika# chmod 711 readfile.sh
localhost:~/Ritika# ./readfile.sh
This is exp 8 of OS Lab.
```

5. WAP to read a file line by line

```
localhost:~/Ritika# cat > car.txt
MG Hector
Grand Vitara
Mercedes Benz
localhost:~/Ritika# vi printfile.sh
#!/bin/bash
myfile="car.txt"
i=1
while read lines; do
    echo "$i: $lines"
    i=$((i+1))
done < "$myfile"

~
~
localhost:~/Ritika# chmod 711 printfile.sh
localhost:~/Ritika# ./printfile.sh
1: MG Hector
2: Grand Vitara
3: Mercedes Benz
```

6. WAP to display system information

```
localhost:~/Ritika# vi system.sh
#!/bin/bash
echo "Date"
date
echo "Uptime"
uptime
echo "Memory usage"
free -m
echo "Network usage"
ip a
~

localhost:~/Ritika# chmod 711 system.sh
localhost:~/Ritika# ./system.sh
Date
Mon Mar 10 18:00:56 UTC 2025
Uptime
 18:00:57 up 5 min,  load average: 0.00, 0.00, 0.00
Memory usage
              total        used        free      shared  buff/cache   available
Mem:            119           4         113           0           1         112
Swap:             0           0           0
Network usage
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN qlen 1000
    link/ether 02:b4:01:49:4e:45 brd ff:ff:ff:ff:ff:ff
```

7. WAP to find and replace text in a string

```
localhost:~/Ritika# vi findreplace.sh
#!/bin/bash
first="I drive BMW and Volvo"
second="Audi"
echo "${first/BMW/$second}"
~

localhost:~/Ritika# chmod u+rwx findreplace.sh
localhost:~/Ritika# ./findreplace.sh
I drive Audi and Volvo
```