OPRATING SYSTEM

Theory Assignment -1

Shell scripting & Linux commands

Name: Onkar Anil Waghmode

Div: SY-CS-D

Roll No: 72

PRN: 12210334

Shellscript

1. Write shell program to check argument is positive or negative

```
#!/bin/bash

if [ $1 -lt 0 ];
then
echo "Negative"
else
echo "Positive"
fi
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./positive_negative.sh 90
Positive
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./positive_negative.sh -25
Negative
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

2. Write arithmetic calculations program by using case structure

```
#!/bin/bash

addition() {
  result=$((num1 + num2))
  echo "Result: $result"
}


subtraction() {
  result=$((num1 - num2))
  echo "Result: $result"
}
```

```
multiplication() {
  result=$((num1 * num2))
  echo "Result: $result"
}

division() {
  if [ $num2 -eq 0 ]; then
    echo "Error: Cannot divide by zero!"
  else
    result=$((num1 / num2 |bc))

    echo "Result: $result"
  fi
}

echo "Enter first number: "
read num1
echo "Enter second number: "
read num2

echo "Select an operation:"

read choice

case $choice in
    +) addition ;;
    -) subtraction ;;
    "*") multiplication ;;
    /) division ;;
    *) echo "Invalid oprator. Please try again." ;;
esac
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./arithmaticopration.sh
Enter first number:
25
Enter second number:
64
Select an operation:
+
Result: 89
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./arithmaticopration.sh
Enter first number:
60
Enter second number:
25
Select an operation:
-
Result: 35
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./arithmaticopration.sh
Enter first number:
25
Enter second number:
25
Select an operation:
*
Result: 625
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./arithmaticopration.sh
Enter first number:
25
Enter second number:
5
Select an operation:
/
Result: 5
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

3. Write program to calculate factorial of given number

```
#!/bin/bash

echo "Enter Number : "
read num

fact=1;

for ((i=$num;i>=1;i--));
do
```

```
fact=$(( $fact * $i ))
done

echo "Factorial : $fact"
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$  ./factorial.sh
Enter Number :
5
Factorial : 120
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$  ./factorial.sh
Enter Number :
7
Factorial : 5040
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

4. Write program to check number is palindrome or not

```
#!/bin/bash

echo "Enter number : "
read num;

temp=$num;

rev=""

while [ $num -gt 0 ]
do
s=$(( $num % 10 ))

num=$(( $num / 10))

rev+=${s}
done

if [ $temp -eq $rev ];
then
```

```
echo "Number is pelindrome"
else
echo "Number is Not pelindrome"
fi
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./pelindrome.sh
Enter number :
1221
Number is pelindrome
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$  ./pelindrome.sh
Enter number :
21231
Number is Not pelindrome
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$  ./pelindrome.sh
Enter number :
122221
Number is pelindrome
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$  ./pelindrome.sh
Enter number :
85647
Number is Not pelindrome
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

5. Write a program to determine the prime number

```
#!/bin/bash

is_prime() {
  num=$1

  if [ $num -lt 2 ]; then
    return 1
  fi

  limit=$(( $num / 2))

  for ((i = 2; i * i <=$limit; i++)); do
    if [ $((num % i)) -eq 0 ]; then
```

```
      return 1
    fi
  done

  return 0
}

echo "Enter Number "
read num

is_prime "$num"

if [ $? -eq 0 ]; then
  echo "$num is a prime number."
else
  echo "$num is not a prime number."
fi
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./primenumber.sh
Enter Number
34
34 is not a prime number.
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./primenumber.sh
Enter Number
33
33 is not a prime number.
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./primenumber.sh
Enter Number
17
17 is a prime number.
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

6. Write a program to reverse of numbers

```
#!/bin/bash

echo "Enter Number : "
read num

rev=""

while [ $num -gt 0 ]
do
e=$(( $num % 10 ))
num=$(( $num / 10 ))
rev+=${e}
done

echo "Reverse string : $rev"
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./reversenumber.sh
Enter Number :
1234
Reverse string : 4321
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./reversenumber.sh
Enter Number :
4532678
Reverse string : 8762354
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

7. Write shell script to check and count occurrence of substring in given string

```
#!/bin/bash

echo "enter String : "
read str

echo "Enter Sub-string : "
read substr

count=0
```

```
ch=${#substr}

limit=$(( ${#str} - $ch))

for (( si=0;si<=$limit;si++ ));
do
check=${str:$si:$ch}
if [ $check == $substr ];
then
count=$(( $count + 1 ))
fi
done

echo "count : $count"
```

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./countsubstring.sh
enter String :
vitpunevitvitpunevitpunevit
Enter Sub-string :
vit
count : 5
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ ./countsubstring.sh
enter String :
vitpunepunevitpunepunevitvitpune
Enter Sub-string :
pune
count : 5
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

Linux Commands

1. Redirect output of ls command to file “out”

```
onkar2004@DESKTOP-U4B25LI:~$ ls > out
onkar2004@DESKTOP-U4B25LI:~$
```

2. Select line in file which has digit one of the character in that line and redirect this output to the file name as “list”

```
onkar2004@DESKTOP-U4B25LI:~$ grep -n "[0-9]" out > list
onkar2004@DESKTOP-U4B25LI:~$
```

3. Assign execute permission to owner and remove read permission

```
onkar2004@DESKTOP-U4B25LI:~$ chmod u+x-r out
onkar2004@DESKTOP-U4B25LI:~$
```

4. Create alias named as “rm” that always delete the file interactively

```
onkar2004@DESKTOP-U4B25LI:~$ alias rm='rm -i'
onkar2004@DESKTOP-U4B25LI:~$ rm out_uppercase
rm: remove regular file 'out_uppercase'? y
onkar2004@DESKTOP-U4B25LI:~$
```

5. Count currently login user to system

```
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$ who | tee logged_in_users.txt | wc -l
0
onkar2004@DESKTOP-U4B25LI:~/shellProgramming$
```

6. List all hidden files under “home” directory

```
onkar2004@DESKTOP-U4B25LI:~$ ls -a | grep "^\."
.
..
.bash_history
.bash_logout
.bashrc
.lesshst
.local
.motd_shown
.profile
.viminfo
onkar2004@DESKTOP-U4B25LI:~$
```

7. Convert lower case files to uppercase of “out” file

```
onkar2004@DESKTOP-U4B25LI:~$ tr '[:lower:]' '[:upper:]' < out > out_uppercase
onkar2004@DESKTOP-U4B25LI:~$
```

8. Display how many times lines are repeated in the given line

```
onkar2004@DESKTOP-U4B25LI:~$ sort out | uniq -c | sort -nr
      1 user_list.txt
      1 ubuntu
      1 shellProgramming
      1 out
      1 ourdate
      1 northeast_states_capitals
      1 northeast_states
      1 mystates_uppercase
      1 mystates
      1 mydate
      1 mycapitals
      1 logged_in_users.txt
      1 list
      1 file2.txt
      1 file1.txt
      1 capitals3
      1 capitals2
      1 capitals1
      1 capitals
      1 OsAssignment1.docs
```