

# Shell Programming

# Practical-4

1. Write a shell script to check whether the entered number is prime or not.

## **Solution:**

1. Numbers divisible by 1 and itself are prime
2. Else
3. Not prime

# Practical-4 (Exercise-1)

- echo Enter number
- read a
- i=2 ,Flag=1
- while [ \$i -lt \$a ]
- do
- x=`expr \$a % \$i`
- if [ \$x -eq 0 ]
- then
- Flag=0
- echo Entered number is not a prime number
- exit
- fi
- i=`expr \$i + 1`
- done
- if [ \$Flag -eq 1 ]
- then
- echo Entered number is a prime number
- fi

# Practical-4

2. Write a shell script to calculate HRA of employees depending upon their basic.

**Solution:**

- read salary and Hra
- $Hra < 100$
- $Hra = Hra * salary / 100$
- Print HRA
- else
- echo Invalid HRA percentage

# Practical-4

- echo Enter the Basic Salary
- read sal
- echo Enter the HRA percentage
- read hra
- if [ \$hra -lt 100 ] then
  - temp=`expr \$hra \\* \$sal`
  - hra=`expr \$temp / 100`
  - echo HRA is \$hra else echo Invalid HRA percentage
- fi

# Practical-4

3. Write a shell script that greets the user by saying Good Morning, Good Afternoon, and Good Evening according to the system time.

**Solution :**

**Command to get current Time**

Hour =\$date +%H or %k (%b ,%c,%F)

Hour <12 Morning ,elif Hour < 18 "Afternoon",else "Evening"

# Practical-4

- `hour=$(date +%H)`
- `if [ $hour -ge 0 -a $hour -lt 12 ] then`
- `greet="Good Morning, $USER"`
- `elif [ $hour -ge 12 -a $hour -lt 18 ] then greet="Good Afternoon, $USER"`  
`else greet="Good evening, $USER"`
- `fi`

# Practical-4

- `$ sh filename.sh argument1 argument2 ... argumentN`
- The `$0` variable
- This holds the name of the script.
- The `$1 $2 ... $N` variables
- These variables hold the arguments provided to the script.
- The `$#` variable
- This variable hold the total number of arguments passed to the script.
- The `$@` and `$*` variables
- They both holds the list of arguments provided to the script.



# Practical-4

- Give the owner rx permissions, but not w
- `chmod u=rx file`
- # Deny rwx permission for group and others
- `chmod go-rwx file`
- # Give write permission to the group
- `chmod g+w file`
- # Give execute permission to everybody
- `chmod a+x file1 file2`
- # OK to combine like this with a comma
- `chmod g+rx,o+x file`

# Practical-4

4. Write a shell script, which takes a filename as command line argument, asks the user if he wants to revoke the read, write permissions for the group and others for that particular file.

If the answer is “y” then it should do so or else, it should abort the operation..

# Practical-4

- `fn=`echo $1``
- `echo "Do you want to revoke the read write permissions`
- `for the group and others press y to do so.."`
- `read ch`
- `if [ $ch = y -o $ch = Y ]`
- `then`
- `temp=`chmod 766 $fn``
- `$temp`
- `fi`

# Practical-4

5. Write a shell script that asks the capital of Gujarat and repeats the question until the user gives correct answer..

# Practical-4

- while [ 1 ]
- **Do**    echo Enter the capital of gujarat
- read cap
- if [ \$cap = "gandhinagar" -o \$cap = "GANDHINAGAR" ]
- Then    echo correct answer
- **Exit**
- fi
- echo wrong answer, try again
- **done**

# Practical-4

6. Write a shell script to display desired line from a file.

# Practical-4

To print 4th line from the file then we will run following commands.

**1.awk** :\$>awk '{if(NR==4) print \$0}' file.txt

**2.sed** :\$>sed -n 4p file.txt

**3.head** :\$>head -n 4 file.txt | tail -n + 4

**OR**

head -\$ln \$fn | tail -1

# Practical-4

- echo Enter file name
- read fn
- echo Enter line number
- read ln
- x=`head -\$ln \$fn | tail -1`
- echo \$x



# Practical-4

7. Write a shell script to count number of newline characters in a file.

```
x=`wc $filename | cut -d " " -f2`
```

```
y=`wc -l $filename
```

# Practical-4

8. Write a shell script to count number of spaces in a file

1. Get no of line
2. Loop for each line
3. Read char of one line at a time
4. Loop until char!=" "
5. If ch=" " count ++
6. Read next char
7. Read next line

# Practical-4

9. Write a Shell script, which counts the number of words in a file, without taking into consideration the blank space, tab spaces and the newline characters using WC.

Solution:

Same as previous Exercise -8

By printing value of count = no of space

## Practical-4

10. Write a Shell script, which counts the number of words in a file, without taking into consideration the blank space, tab spaces and the newline characters without using WC.

### Solution:

```
x=`wc $fn|cut -d " " -f3`
```

# Practical-4

11. Write a Shell script, which counts the number of characters in a file, without taking into consideration the blank space, tab spaces and the newline characters using WC.

## Solution:

```
x=`wc $fn|cut -d " " -f3`
```

## Practical-4

12. Write a Shell script, which counts the number of characters in a file, without taking into consideration the blank space, tab spaces and the newline characters without using WC.

### Solution:

```
x=`wc $fn|cut -d " " -f3`
```