ASSIGNMENT 2

Write shell scripts for following:

1. Check whether the number is positive or negative

**Ans :**

**Code:**

echo "enter the number"

read a

if [ $a -lt 0 ]

then

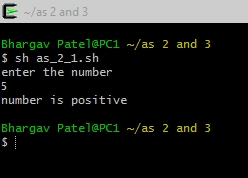
echo "number is negitive"

else

echo "number is positive"

fi

**OutPut:**

****

2. Print a menu and ask user to enter the 1 or 2 as per their favorite OS .

Menu

1. Linux

2. Windows

Select your OS Choice (1/2).

Print appropriate message otherwise.

**Ans:**

**Code:**

echo -e "\e[4mMenu\e[0m"

echo "1. Linux"

echo "2. Windows"

echo "Select your OS choice (1/2)."

read a

case $a in

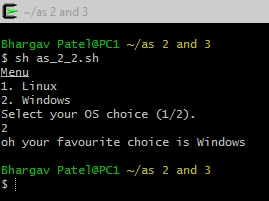
1) echo "oh your favourite choice is Linux";;

2) echo "oh your favourite choice is Windows";;

\*) echo "wrong choice selected!!";;

esac

**Output:**

****

3. Check whether the number is positive, negative or zero.

**Ans:**

**Code:**

echo "enter the number"

read a

if [ $a -lt 0 ]

then

echo "Number you entered is Negitive"

elif [ $a -gt 0 ]

then

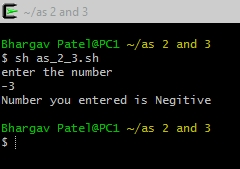
echo "Number you entered is Positive"

else

echo "Number you entered is zero"

fi

**Output:**

****

4. Print multiplication table

**Ans:**

**Code:**

echo "MULTIPLICATION TABLE"

echo "enter the number to show the table of"

read a

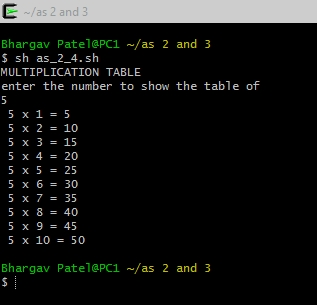
for i in {1..10}

do

echo " $a x $i = `expr $a \\* $i`"

done

**Output:**

****

5. Print the following (if input is 5 for example)

1 1 1 1 1

2 2 2 2 2

3 3 3 3 3

4 4 4 4 4

5 5 5 5 5

**Ans:**

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

while [ $j -le $a ]

do

echo -n "$i "

j=`expr $j + 1`

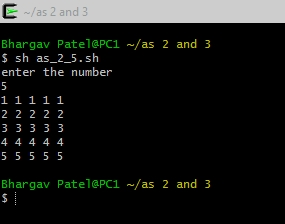
done

echo

i=`expr $i + 1`

done

**Output:**

****

6. Chessboard

**Ans:**

**Code:**

echo "enter the width of chess board"

read a

i=0

while [ $i -le $a ]

do

j=0

while [ $j -le $a ]

do

b=`expr $i + $j`

b=`expr $b % 2`

if [ $b -eq 0 ]

then

echo -e -n "\033[47m "

else

echo -e -n "\033[40m "

fi

j=`expr $j + 1`

done

echo -e -n "\033[40m "

echo ""

i=`expr $i + 1`

done

**Output:**



7. Ask user to enter vehicle type and print appropriate message for rental

1. car

2. Van

3. Bicycle

Default: I cannot get rental for your vehicle

**Ans:**

**Code:**

echo "enter your vehicle type"

read a

case "$a" in

"Car") echo "yes i can get your car on rental";;

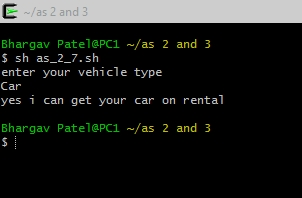
"Bicycle") echo "yes i can get your Bicycle on rental";;

"Van") echo "yes i can get your Van on rental";;

\*) echo "no i can not get your vehicle on rental";;

esac

**Output:**

****

ASSIGNMENT 3

Write shell scripts for following:

1. Addition of two real numbers.

**Ans:**

**Code:**

echo "enter num 1"

read a

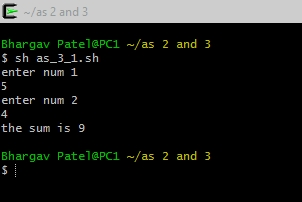
echo "enter num 2"

read b

c=`expr $a + $b`

echo "the sum is $c"

**Output:**

****

2. Write script to determine whether given file or directory exist or not, file name is supplied as

command line argument, also check for sufficient number of command line argument.

**Ans:**

**Code:**

PASSED=$1

if [ $# -ne 1 ]

then

echo "more then 1 or zero arguments are not allowed!!"

else

if [ -d "${PASSED}" ]

then

echo "$PASSED is a directory and it is exist"

else

if [ -f "${PASSED}" ]

then

echo "${PASSED} is a file and it is exist"

else

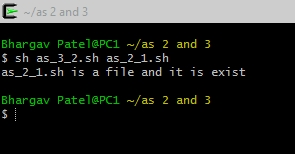
echo "${PASSED} is not exist"

fi

fi

fi

**Output:**

****

3. Write script to implement getopts statement, your script should understand command line

argument with following options:

-c clear the screen

-d show list of files in current working directory

-e exit

**Ans:**

**Code:**

cls()

{

clear

echo "Clear screen, press a key . . ."

read

return

}

show\_ls()

{

ls

echo "list files, press a key . . ."

read

return

}

if [ $# -eq 0 ]

then

echo "you entered zero argument"

exit 1

fi

while getopts cdme: opt

do

case "$opt" in

c) cls;;

d) show\_ls;;

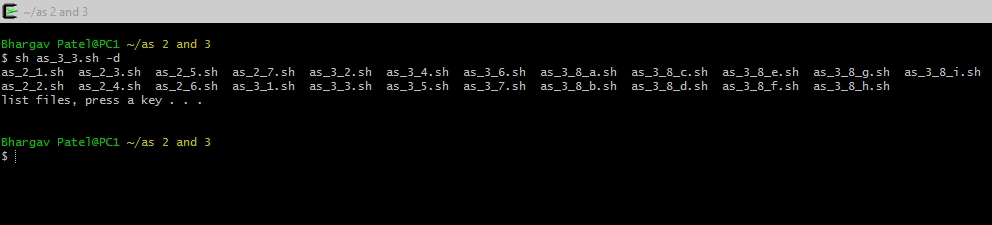
e) exit 1;;

\?) echo "enter valid arguments"

esac

done

**Output:**

****

4. Write a script called sayHello which prints “Good Morning”, Good Afternoon” or “good

Evening” as per the time of the system.

**Ans:**

**Code:**

a=`date '+%H'`

if [ $a -lt 12 ]

then

echo "Good Morning Bhaggu"

elif [ $a -lt 18 ]

then

echo "Good Afternoon Bhaggu"

elif [ $a -lt 22 ]

then

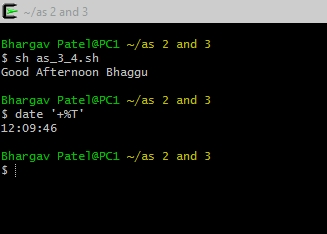
echo "Good Evening Bhaggu"

else

echo "Good Night Bhaggu"

fi

**Output:**



5. Write a script, that will print, Message &quot;Hello World&quot; , in Bold and Blink effect, and in different colors like red, brown etc using echo command.

**Ans:**

**Code:**

clear

while [ 0 -eq 0 ]

do

for i in {0..7}

do

tput bold blink

tput setaf $i

tput cup 15 75

echo -n "Hello World!"

sleep 0.6

done

done

**Output:**

****

6. Write a script to search for a pattern in a file using GREP

**Ans:**

**Code:**

echo "Enter file name..."

read fname;

if [ -f $fname ]

then

echo "Enter the search pattern"

read pattern

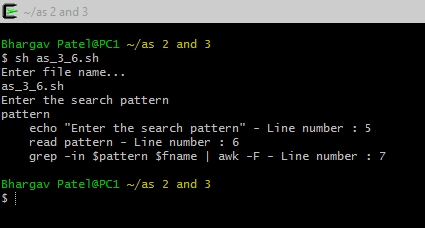
grep -in $pattern $fname | awk -F: '{print $2" - Line number : "$1}'

else

echo "enter valid file name"

fi

**Output:**

****

7. Write a script to print “String has been Found” or “String has not been found” after

searching for a pattern in a file using GREP

**Ans:**

**Code:**

echo "Enter file name..."

read fname;

if [ -f $fname ]

then

echo "Enter the search pattern"

read pattern

co=`grep -c $pattern $fname`

if [ $co -gt 0 ]

then

echo "string is found"

else

echo "No String is not found"

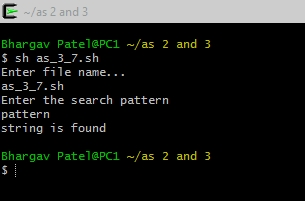
fi

else

echo "enter valid file name"

fi

**Output:**

****

8. Write shell script using for loop to print the following patterns on screen

a. For2

1

22

333

**Ans:**

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

echo -n "$i "

j=`expr $j + 1`

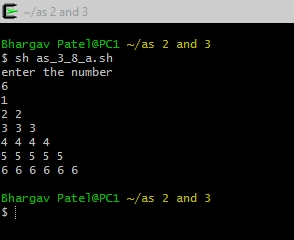
done

echo

i=`expr $i + 1`

done

**Output:**

****

b. For3

1

1 2

1 2 3

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

echo -n "$j "

j=`expr $j + 1`

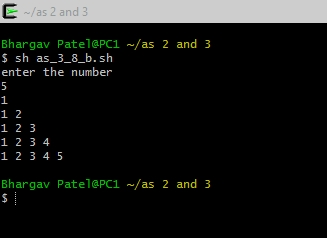
done

echo

i=`expr $i + 1`

done

**Output:**



c. for 4

|\_

| |\_

| | |\_

**Code:**

echo "enter the number"

read a

a=`expr $a \\* 2`

i=2

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

if [ $j -eq $i ]

then

echo -n "\_"

elif [ `expr $j % 2` -eq 0 ]

then

echo -n " "

else

echo -n "|"

fi

j=`expr $j + 1`

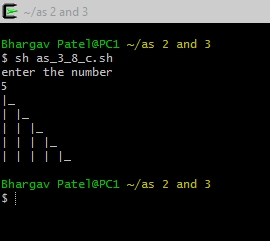
done

echo

i=`expr $i + 2`

done

**Output:**

****

**d. for 5**

**\***

**\* \***

**\* \* \***

**Code:**

echo "enter the number"

read a

a=`expr $a \\* 2`

i=1

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

echo -n "\*"

j=`expr $j + 1`

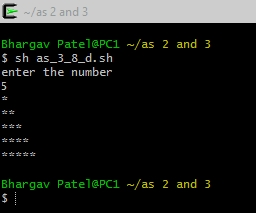
done

echo

i=`expr $i + 1`

done

**Output:**



e. for 7

LINUX\_\_

LINUXLINUX\_\_

LINUXLINUX\_\_

LINUX\_\_

**Ans:**

echo "enter the number"

read a

str="LINUX"

i=1

a=`expr $a / 2`

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

echo -n "$str"

j=`expr $j + 1`

done

j=1

while [ $j -le $a ]

do

echo -n "\_"

j=`expr $j + 1`

done

echo

i=`expr $i + 1`

done

i=$a

while [ $i -ge 1 ]

do

j=1

while [ $j -le $i ]

do

echo -n "$str"

j=`expr $j + 1`

done

j=1

while [ $j -le $a ]

do

echo -n "\_"

j=`expr $j + 1`

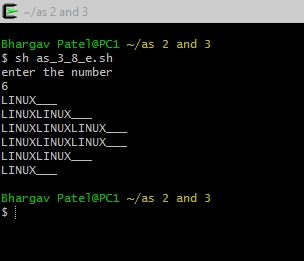
done

echo

i=`expr $i - 1`

done

**Output:**



f. for 6

\*

\* \*

\* \*

\*

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

while [ $j -le $i ]

do

echo -n "\*"

j=`expr $j + 1`

done

echo

i=`expr $i + 1`

done

i=$a

while [ $i -ge 1 ]

do

j=1

while [ $j -le $i ]

do

echo -n "\*"

j=`expr $j + 1`

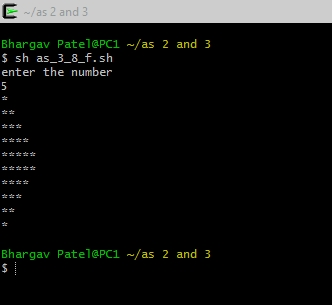
done

echo

i=`expr $i - 1`

done

**Out Put:**

****

**g. for 8**

**1**

**2 2**

**3 3 3**

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

#c=`expr $a / 2`

#c=$a

x=`expr $a - $i`

while [ $j -le $x ]

do

echo -n " "

j=`expr $j + 1`

done

j=1

while [ $j -le $i ]

do

echo -n "$i "

j=`expr $j + 1`

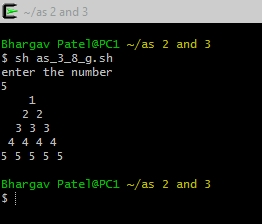
done

echo

i=`expr $i + 1`

done

**Output:**

****

h. for 8

\*

\* \*

\* \* \*

Code:

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

#c=`expr $a / 2`

#c=$a

x=`expr $a - $i`

while [ $j -le $x ]

do

echo -n " "

j=`expr $j + 1`

done

j=1

while [ $j -le $i ]

do

echo -n "\* "

j=`expr $j + 1`

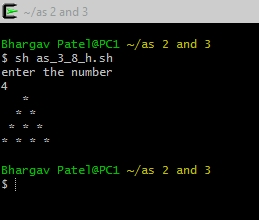
done

echo

i=`expr $i + 1`

done

OutPut:



**I. for 9**

**Code:**

echo "enter the number"

read a

i=1

while [ $i -le $a ]

do

j=1

x=`expr $a - $i`

while [ $j -le $x ]

do

echo -n " "

j=`expr $j + 1`

done

j=1

while [ $j -le $i ]

do

echo -n "\* "

j=`expr $j + 1`

done

echo

i=`expr $i + 1`

done

i=$a

while [ $i -ge 1 ]

do

j=1

x=`expr $a - $i`

while [ $j -le $x ]

do

echo -n " "

j=`expr $j + 1`

done

j=1

while [ $j -le $i ]

do

echo -n "\* "

j=`expr $j + 1`

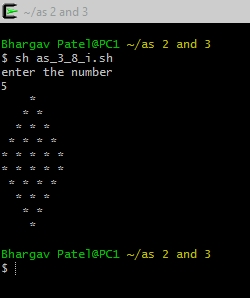
done

echo

i=`expr $i - 1`

done

**Out Put:**

****