DIKSHITHA VANI V
205001032

SSN COLLEGE OF ENGINEERING, KALAVAKKAM
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Simple Programs using shell scripts

1. Write a shell script that prints “Shell Scripting is Fun!” on the screen. Modify the shell script above to include a variable. The variable will hold the contents of the message “Shell Scripting is Fun!”

```
gml29:~ csea29$ cat>assig7.sh
var=\”Shell Scripting is fun!\”
echo $var
^C
```

bash-3.2$ bash assig7.sh
“Shell Scripting is fun!”

2. Print the values of the environment variables HOME, USER, SHELL and PATH with set, print env and echo.

```
bash-3.2$ cat>assig7.sh
echo $HOME
echo $USER
echo $SHELL
echo $PATH
^C
```

bash-3.2$ bash assig7.sh
/Users/csea29
csea29
/bin/bash
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin

```
bash-3.2$ cat>assig7.sh
set|grep HOME
set|grep USER
set|grep SHELL
set|grep PATH
^C
```

```
bash-3.2$ bash assig7.sh
HOME=/Users/csea29
USER=csea29
__CF_USER_TEXT_ENCODING=0x1FA:0x0:0x0
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
bash-3.2$ cat>assig7.sh
set|grep HOME
set|grep ^USER
set|grep ^SHELL
set|grep PATH
^C
```

```
bash-3.2$ bash assig7.sh
HOME=/Users/csea29
USER=csea29
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
PATH=/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin
```

```
bash-3.2$ cat>assig7.sh
printenv HOME
printenv USER
printenv SHELL
printenv PATH
```

bash-3.2$ bash assig7.sh
/Users/csea29
csea29
/bin/bash
/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin

3.Store the output of the command “hostname” in a variable. Display “This script is running on _.” where “_” is the output of the “hostname” command.

```
bash-3.2$ cat>assig7.sh
var=$HOSTNAME
echo "This script is running on $var"
```

bash-3.2$ bash assig7.sh
This script is running on gml29.local

4. Get two numbers a and b from user using read statement. Do the following:
a. Add the two numbers
b. Subtract the numbers
c. Multiply the numbers
d. Divide the numbers
Print the result.

```
bash-3.2$ cat>assig7.sh
echo "Enter two numbers"
read a b
echo "sum is:$(($a+$b))"
echo "diff is:$(($a-$b))"
echo "product is:$(($a*$b))"
echo "quotient is:$(($a/$b))"
^C
```

bash-3.2$ bash assig7.sh
Enter two numbers
10 5
sum is:15

diff is:5
product is:50
quotient is:2
bash-3.2$

5. Get length and breadth for a rectangle and radius for a circle using command line argument. Calculate area and perimeter of the rectangle and also area and circumference of a Use the special character data types and display the arguments using them

```
bash-3.2$ cat>assig7.sh
echo "Area of rectangle:" && echo $1*$2|bc
echo "Perimeter of rectangle:" && echo "2*($1+$2)"|bc
echo "Area of circle:" && echo 3.14*$3*$3|bc
echo "circumference of circle:" && echo 2*3.14*$3|bc
```

bash-3.2$ bash dat.sh 10 20 5
Area of rectangle:
200
Perimeter of rectangle:
60
Area of circle:
78.50
circumference of circle:
31.40

6. Temperature of a city in Fahrenheit degree is input through the keyboard. Write a program to convert this temperature into Centigrade degrees.
Formula is
c = (f - 32)*5 /9
f=9/5*c+32

```
bachi@dikshu:~$ cat>scr.sh
```

```
echo Enter temperature in F
read f
c=$(((f-32)*5/9))
echo Temperature in Celsius: $c
```

bachi@dikshu:~$ bash scr.sh
Enter temperature in F
32
Temperature in Celsius: 0

7. Write a shell script to calculate the net salary of an employee in a particular month
considering various allowances (TA, DA, HRA) and deductions (INCOME TAX, PROVIDEND
FUND) as:
a. TA=15 percent of basic salary
b. DA=2 percent of basic salary
c. HRA=10 percent of basic salary
d. INCOME TAX=5 percent of salary
e. PROVIDEND FUND=10 percent of salary

```
gml29:~ cse29$ cat>employee.sh
read basic
TA=$(echo "0.15*$basic"|bc)
DA=$(echo "0.02*$basic"|bc)
HRA=$(echo "0.1*$basic"|bc)
INCOME_TAX=$(echo "0.05*basic"|bc)
PROVIDENT_FUND=$(echo "0.1*basic"|bc)
Salary=$(echo "$basic+$HRA+$DA-$INCOME_TAX-$PROVIDENT_FUND"|bc)
echo $Salary
```

gml29:~ cse29$ bash employee.sh
100000
112000.00
gml29:~ cse29$ bash employee.sh
10000

11200.00

8. In a town, the percentage of men is 52. Rest all are women. The percentage of total literacy is 48. If total percentage of literate men is 35 of the total population, WAP to find the total number of illiterate men and women. The population of the town is 80,000.

```
gml29:~ cse29$ cat>pop.sh
pop=80000
men=$(echo "0.52*$pop"|bc)
men_literate=$(echo "0.35*$pop"|bc)
total_literate=$(echo "0.48*$pop"|bc)
women_literate=$(echo "$total_lierate-$men_literate"|bc)
men_illiterate=$(echo "$men-$men_literate"|bc)
women_illiterate=$(echo "$pop-$total_literate-$men_illiterate"|bc)
total_illiterate=$(echo "$pop-$total_literate"|bc)
echo $total_illiterate
echo $women_illiterate
echo $men_illiterate
```

gml29:~ cse29$ bash pop.sh
41600.00
28000.00
13600.00

9. Write a shell script that displays “man”,”bear”,”pig”,”dog”,”cat”,and “sheep” on the screen with each appearing on a separate line. Use special characters to display the filename, no of parameters, display the arguments each on one line, use appropriate command to dispaly the differences between $@, $*. Explain how about the status code of the script.

```
bachi@dikshu:~$ cat>ex7.sh
```

```
echo "$1"
echo "$2"
echo "$3"
echo "$4"
echo "$5"
echo "$6"
echo "FILE NAME:$0"
echo "NO OF PARAMETERS:$#"
echo "ARGUMENTS: USING \$@"
for i in "$@"; do echo $i; done
echo "ARGUMENTS: USING \$#"
for i in "$*"; do echo $i; done
```

bachi@dikshu:~$ bash ex7.sh man bear pig dog cat sheep
man
bear
pig
dog
cat
sheep
FILE NAME:ex7.sh
NO OF PARAMETERS:6
ARGUMENTS: USING $@
man
bear
pig
dog
cat
sheep
ARGUMENTS: USING $#
man bear pig dog cat sheep

10. Write a shell script that prompts the user for a name of a file or directory and reports if it is a regular file, a directory, or another type of file. Also perform an ls command against the file or directory with the long listing option.

```
bachi@dikshu:~$ cat>ex7.sh
read -p "Enter file name:" filename
if [ -e $filename ]; then
   if [ -d $filename ]
   then
         echo "Its a directory"
         ls -l $filename

   elif [ -f $filename ]
   then
         echo "Its a regular file"

   elif [ -c $filename ]
   then
         echo "Its a character special file"
   elif [-b $filename ]
   then
         echo "Its a block special file"
   fi

else
  echo "FILE DOESNT EXIST"
fi
```

bachi@dikshu:~$ bash ex7.sh
Enter file name:hello
Its a regular file

bachi@dikshu:~$ bash ex7.sh
Enter file name:tello
FILE DOESNT EXIST

11. Modify the previous script to that it accepts unlimited number of files and directories as arguments and display the information about it. (use cat for files and ls-l for directories)

```
bachi@dikshu:~$ cat ex7.sh
file=$@
for f in $file
do
  if [ -e $f ]; then
        if [ -d $f ]
        then
        echo "Its a directory"
        ls -l $f

        elif [ -f $f ]
        then
        echo "Its a regular file"

        elif [ -c $f ]
        then
        echo "Its a character special file"
        elif [-b $f ]
        then
        echo "Its a block special file"
        fi

  else
   echo "FILE DOESNT EXIST"
  fi
done
```

bachi@dikshu:~$ bash ex7.sh hello tello cello
Its a regular file
FILE DOESNT EXIST
FILE DOESNT EXIST

```
bachi@dikshu:~$ ls *ello
Hello

bachi@dikshu:~$ ls -l hello
-rw-rw-r-- 1 bachi bachi 141 Dec  5 11:28 hello
bachi@dikshu:~$ ls -l cello
ls: cannot access 'cello': No such file or directory
bachi@dikshu:~$ ls -l tello
ls: cannot access 'tello': No such file or directory
bachi@dikshu:~$
```

12. Write a shell script to display the current date and cut down the month of the date and store it in the file date.txt. Use ` ` in the command to store the content in the file and display the file. Also use an alias function to cut down the day of the week and execute the command.

```
bachi@dikshu:~$ cat dat.sh
date
date|cut -d' ' -f2>date.txt
echo "MONTH: "
cat date.txt
shopt -s expand_aliases
alias week_day="date|cut -d' ' -f1"
week_day

bachi@dikshu:~$ bash dat.sh
Sun Dec 12 09:18:54 IST 2021
MONTH:
Dec
Sun
```

13. Create the following files and change the permissions specified
File1 701
File2 400
File3 300
File4 676
File5 045
File6 177
File7 234
File8 507
Write a shell script to find the number of readable, writable and executable files.

```
bachi@dikshu:~$ cat ex7.sh
touch File1
touch File2
touch File3
touch File4
touch File5
touch File6
touch File7
touch File8
chmod 701 File1
chmod 400 File2
chmod 300 File3
chmod 676 File4
chmod 045 File5
chmod 177 File6
chmod 234 File7
chmod 507 File8
for i in File*
do
 if [ -r $i ]
 then
   echo "$i is readable"
 else
   echo "$i is not readable"
 fi
```

```
if [ -w $i ]
 then
   echo "$i is writable"
 else
   echo "$i is not writable"
 fi

 if [ -x $i ]
 then
   echo "$i is executable"
 else
   echo "$i is not executable"
 fi
echo
done
```

bachi@dikshu:~$ bash ex7.sh
File1 is readable
File1 is writable
File1 is executable

File2 is readable
File2 is not writable
File2 is not executable

File3 is not readable
File3 is writable
File3 is executable

File4 is readable
File4 is writable
File4 is not executable

File5 is not readable
File5 is not writable
File5 is not executable

File6 is not readable
File6 is not writable
File6 is executable

File7 is not readable
File7 is writable
File7 is not executable

File8 is readable
File8 is not writable
File8 is executable

```
bachi@dikshu:~$ bash ex7.sh
bachi@dikshu:~$ ls  -l File*
-rwx-----x 1 bachi bachi 0 Dec 12 08:59 File1
-r-------- 1 bachi bachi 0 Dec 12 08:59 File2
--wx------ 1 bachi bachi 0 Dec 12 08:59 File3
-rw-rwxrw- 1 bachi bachi 0 Dec 12 08:59 File4
----r--r-x 1 bachi bachi 0 Dec 12 08:59 File5
---xrwxrwx 1 bachi bachi 0 Dec 12 08:59 File6
--w--wxr-- 1 bachi bachi 0 Dec 12 08:59 File7
-r-x---rwx 1 bachi bachi 0 Dec 12 08:59 File8
```

13. Write the shell script that renames all files in the current directory that end in ".jpg" to begin with today's date in the following format: YYYY-MM-DD. For example, if a picture of my cat was in the current directory and today was October 31,2016 it would change name from "mycat.jpg" to "2016–10–31-mycat.jpg".

```
bachi@dikshu:~$ cat ex7.sh
day=`date +%F`
```

```
cd ~/pic
for file in *.jpg
do
mv $file ${day}-${file}
done
```

14. Write a script that executes the command “cat/etc/shadow”. If the command return a 0 exit status,report “command succeeded” and exit with a 0 exit status. If the command returns a non-zero exit status, report “Command failed” and exit with a 1 exit status.

```
bachi@dikshu:~$ cat ex7.sh
cat /etc/shadow
if [ "$?" -eq "0" ]
then
echo "Command succeeded"
exit 0
else
echo "Command failed"
exit 1
fi
```

bachi@dikshu:~$ bash ex7.sh
cat: /etc/shadow: Permission denied
Command failed