COLLEGE OF ENGINEERING

(Autonomous Institution)

ESTD. 1949

RECORD OF PRACTICAL WORK

NAME : Akshaya Deep Prasad

SUBJECT : UNIX Shell Scripts

SEMESTER : V BRANCH : CSE

ROLLNO : USN : 1BM19CS013

B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

RECORD OF PRACTICAL WORK

NAME : Akshaya Deep Prasad

SUBJECT : Unix Shell Scripts

SEMESTER : V BRANCH : CSE

ROLLNO : USN : 1BM19CS013

B M S. COLLEGE OF ENGINEERING

(Autonomous Institution)

LABORATORY CERTIFICATE

This is to Certify that Mr. / Ms. Akshaya Deep Prasad

has Satisfactorily completed the course of experiments in Practical

Unix Shell Scripts Prescribed

by the Visvesvaraya Technological University for Fifth

Semester CSE; B.E. Course in the Laboratory of the college

in the year 2021 - 20

Head of the Department

Staff incharge of the Batch

Date : .....

Marks	
Maximum	Obtained

Name of the Candidate : Akshaya Deep Prasad

Roll No : ..... USN : 1BM19CS013

Akshaya

Signature of the Candidate

Date	Experiment	Marks Obtained	Page No.
25-10-21	Shell script to check if a year is leap or not		
25-10-21	Shell script to find area of a circle		
25-10-21	Shell script to check if a number is positive, negative or zero		
25-10-21	Shell script to find biggest of three numbers.	8/11/21	
08-11-20	Shell script to find factorial of a number		
08-11-21	Shell script to compute gross salary of an employee		
08-11-21	Shell script to convert Fahrenheit to Celcius		
08-11-21	Shell script to perform arithmetic operations on two numbers	15/11/21	
15-11-21	Shell script to find sum of even numbers upto n		
15-11-21	Shell script to print combinations of 123		
15-11-21	Shell script to find power of a number.		

Particulars of the Experiments Performed

CONTENTS

Date 25-10-21

Expt. No. 1

Page No.

Shell script to check if the given year is leap or not.

```
#!/bin/bash
echo "Enter year"
read leap
if [ $((( leap %.400)) -eq 0 ]
then
 echo "Leap year"
elif [ $((leap %.100 )) -eq 0 ]
then
echo "Not a leap year"
elif [ $((leap %.4 )) -eq 0 ]
then
echo "Leap year"
else
echo "Not a leap year"
fi
```

Output:

Enter year	Enter year
2020	~~2100~~ 2100
Leap year	Not a leap year.

15/21

Teacher's Signature : _____

Date 25/10/21

Expt. No. 2 Page No.

Q: Shell script to find area of a circle.

```
#!/bin/bash
echo "Enter radius of circle"
pi=3.142
echo "Area of circle is :"
area=`echo "$pi \* $r \* $r | bc"`
echo $area
```

Output:

Enter radius of circle
7
Area of the circle is:
153.958

N
25/10/21

Teacher's Signature :

Date: 25-10-21

Expt. No. 3 Page No.

Q.4 Shell script to check whether a number is zero, positive or negative.

```
#!/bin/sh
echo "Enter a number"
read num
if [ $no -eq 0 ]
then
echo "The number is zero"
elif [ $no -lt 0 ]
then
echo "The number is Negative"
else
echo "The number is positive"
fi
```

Output:

```
Enter a number
9
The number is positive.
```

25/10/21

Teacher's Signature :

Date 25-10-21

Expt. No. 4 Page No.

Q. Shell script to find biggest of three numbers

```
#!/bin/sh
echo "Enter first number"
read a
echo "Enter second number"
read b
echo "Enter third number"
read c
if [$a -gt $b -a $a -gt $c]
then
echo "$a is the biggest number"
elif [$b -gt $a -a $b -gt $c]
then
echo "$b is the biggest number"
else
echo "$c is the biggest number"
fi
```

Output:

```
Enter first number
17
Enter second number
10
Enter third number
23
23 is the biggest number.
```

Teacher's Signature :

Date 08-11-21

Expt. No. 5 Page No.

Q.6 Shell script to find factorial of a number.

```
#! /bin/bash
echo "Enter a number"
read num
fact=1
while [ $num -gt 1 ]
do
    fact= $(( fact * num ))
    num = $ ((num-1))
done
$ echo $ fact
```

Output:

```
Enter a number
7
5040
```

Teacher's Signature : _____

Date 08-11-21

Expt. No. 6 Page No.

Q.6 Shell script to compute gross salary of an employee.

```
#! /bin/sh
echo "Enter the Basic Salary"
read basic
gross = $ ((basic +((basic /100)*20)+ (basic/100) *10))

echo " Gross Salary is : $ gross"
```

Output:

```
Enter the basic salary
50000
Gross salary is : 65000
```

8/11/21

Teacher's Signature : _____

Expt. No. 7 Page No.

Q.6 Shell script to convert temperature in Farhenheit to Celcius.

```
#! /bin/sh
echo "Enter temperature in Farhenheit"
read f
c= $( echo "scale=2; (5/9)* ($f-32 )" )
echo "$f Farhenheit = $ C Celcius"
```

Output:

```
Enter temperature in Farhenheit
212
212 Farhenheit = 100 Celcius
```

8/11/21

Teacher's Signature : _____

Output :

Enter two numbers
9
3
1: Addition
2: Subtraction
3: Multiplication
4: Division
Enter Choice
4 ↵

1 ↵	2 ↵	3 ↵	10
12	6	27	Invalid Choice

3

Date 08-11-21

Expt. No. 8

Page No.

Q4 Shell script to ~~find~~ perform arithmetic operation on 2 numbers.

```
#!/bin/sh
echo "Enter two numbers"
read a
read b
echo "1: Addition \n 2: Subtraction \n 3: Multiplication \n
      4: Division \n Enter Choice "
read choice
case $(choice) in
    1) echo "$((a+b))
       ;;
    2) echo "$((a-b))"
       ;;
    3) echo "$((a*b))"
       ;;
    4) echo "$((a/b))"
       ;;
    *) echo "Invalid Choice"
       ;;
```

Teacher's Signature : _____

```
#!/bin/bash
echo "Enter a number"
read n
i=2
while [ $i -lt $n ]
do
        sum = $((sum +i))
        i = $((i+2))
done
echo "$sum"
```

Output:

```
Enter a number
14
42
```

Teacher's Signature : _____

Date 08-11-21

Expt. No. 10 Page No.

Q.4 Shell script to print combinations of 1, 2, 3.

```
#!/bin/bash
for i in 1 2 3
do
    for j in 1 2 3
    do
        for k in 1 2 3
        do
            echo "$i $j $k"
        done
    done
done
```

Output

1 1 1	1 3 1	2 2 1	3 1 1	3 3 1
1 1 2	1 3 2	2 2 2	3 1 2	3 3 2
1 1 3	1 3 3	2 2 3	3 1 3	3 3 3
1 2 1	2 1 1	2 3 1	3 2 1	
1 2 2	2 1 2	2 3 2	3 2 2	
1 2 3	2 1 3	2 3 3	3 2 3	

12

8/11/21

Teacher's Signature : _____

Output :

Enter a number
3
Enter power
4
81

Date 08-11-21

Expt. No. 11

Page No.

Q) Shell script to find power of a number

```
#!/bin/bash
read n
echo "Enter power"
read pow
count=0
res=1
while [ $pow -ne $count ]
do
        res = $((res *n))
        count = $((count +1))
done
echo "$res"
```

8/11/21

Teacher's Signature : _____

Output:

Enter a number
11
66

Date 05/11/21

Expt. No. 12

Page No.

Q.1 Shell script to find sum of n natural numbers.

```
#!/bin/bash
echo "Enter a number"
read n
res=0
while [$n -gt 0]
do
        res=$((res+n))
        n=$((n-1))
done
echo "$res"
```

NB 8/11/21

Teacher's Signature : _____

Lab - 13

```
pass=6
for((i=0;i<6;i++))
do
    echo "Enter Subject :"
read sub
echo "Enter CIE marks out of 100 :"
read cie
echo "Enter SEE marksout of 100"
read see
 cie= $((cie/2))
  se= $((see/2))
  tot= $((cie+see))
echo $tot
case $tot in
100) echo "$grade";;
9[0-9]) echo "S grade";;
 8[0-9]) echo "A grade";;
 7[0-9]) echo "B grade";;
 6[0-9]) echo "C grade";;
 5[0-9]) echo "D grade";;
 4[0-9] echo "E grade";;
 [0-3][0-9]) echo "Fail in subject"
                pass=$((pass-1));;
 *) echo "Enter valid marks :"
 esac

done
echo "Total pass is $pass"
fail=$((6-pass))
echo "Total fail is $fail"
```

Lab-14

Find fibonacci series upto n.

```
echo "Enter a number"
read n
a=0
b=1
c=2
d=0
echo -e "$a $b \c"
while [$c -lt $n]
do
    c=`expr $c+1`
    d=`expr $a + $b`
    echo -e "$d \c"
    a=$b
    b=$d
done.
```

Output

```
Enter a number 5
0
1
1
2
3
```

Lab - 15

Count no. of vowels in a string:

```
echo "Enter the string"
read S
count=0
len=`expr "$S" : '.*'`
for ((i=1; i<=len; i++))
do
    c=`echo $S | cut -c $i`
    case $c in
    [aeiouAEIOU]) count=$((count+1))
    esac
done
echo "Number of vowels is $count"
```

Output:

```
Enter the string: Hello
Number of vowels is 2
```

Lab - 16

Find no. of lines, characters and words in a file :-

Sol^n :
```
echo "Enter file to open: "
read f
lines = `wc -l < $f`
words = `wc -w < $f`
characters = `wc -m < $f`
echo "Lines = $lines \n Words = $words \n Characters = $characters"
```

Output:

```
Enter file to open: Lab16.sh
lines = 6
words = 34
characters = 145
```

Lab-17

Q:- Program to output contents of environment list

Soln:
```
#include <stdio.h>
int main ( int argc, char *argv [])
{
    int i;
    char **ptr;
    extern char **environment;
    for (ptr = environment; ptr != 0; ptr++)
        printf ("%s\n", ptr);
    return 0;
}
```

Note: Output is system dependent.

Lab -18

Q-6 Write a program to emulate ln command.

Soln.
```
# include < stdio.h>
# include < sys/types.h>
# include < unistd.h>
# include < string.h>
int main (int argc, char *argv[]) {
    if (argu< 3|| argu>4 || (argu==4 && strcmp (argu[1], "-s")))
    {
        print ("Usage: /a.out [-s] <org-file> new-links
                >/n );
        return 1;
    }
    if (argc==4 ){
        if ((syslink (argv[2], argv[1] ))==-1)
            printf ("Cannot create symbolic link \n");
        else printf ("Symbolic link created \n");

    }
    else {
        if ((link (argu[3], argv[2] ))==-1)
        printf ("Cannot create symbolic link \n");

    } else printf ("Hard link created\n");

    }
    return 0;

}
```

Lab-19

Q.) C program to print POSIX defined print functions configuration using test matrices:

Sol^{n}:

```
#define _POSIX_SOURCE
#define _POSIX_C_SOURCE 199309L
#include <stdio.h>
#include <unistd.h>
int main(){
  #ifdef _POSIX_JOB_CONTROL
     printf("System supports Job control\n");
  #else printf("System does not support Job control\n");
  #endif
  #ifdef _POSIX_SAVED_IDS
     printf("System supports saved set-UID and saved set-GID\n");
  #else
     printf("System does not support saved UID\n");
  #endif
  #ifdef _POSIX_CHOWN_RESTRICTED
     printf("Chown Restricted option is %d\n",_POSIX_CHOWN_RESTRICTED);
  #else printf("System does not support Chown Restricted option\n");
  endif
  #ifdef _POSIX_VDISABLE
   printf("Disable character for terminal files is %d\n"
                                        _POSIX_VDISABLE");
  else
    printf("System does not support VDISABLE\n");
  endif
  return 0;
  }
```

Output:

Supports Job control
Supports saved set-UID and saved set-GID
Chown Restricted option is 1.
Pathname trunc option is 1.
Disable characters for terminal file is 0.

Lab-20

Q. Program to demonstrate interprocess communication between reader process and writer process.

Sol n:

```
#include <sys/types.h>
#include <unistd.h>
#include <fcntl.h>
#include <sys/stat.h>
#include <string.h>
#include <errno.h>
#include <stdio.h>

int main (int argc, char * argv[]) {

    int fd;
    char buf[256];
    if (argc != 3 && argc != 2) {
        printf ("Usage %s <fifo> [<arg>] \n", argv[0]);
        return 0; }
    mkfifo (argv[1], S_IFIFO | S_IRWXU | S_IRWXG
                    | S_IRWXO);
    if (argc == 2) {
        fd = open (argv[1], O_RDONLY | O_NONBLOCK);
        while (read (fd, buf, sizeof (buf)) > 0)
            printf ("%s", buf);

    }
    else {
        fd = open (argv[1], O_WRONLY);

        write (fd, argv[2], strlen (argv[2]));

    }
    close (fd);

}
```

Output:

```
./a.out FIFO | "This is USP Lab"
./a.out FIFO // Terminal 2
This is USP Lab.
```