Particulars of the Experiments Performed

CONTENTS

Particulars of the Experiments Performed

CONTENTS

O/P

Enter year : 700

Not Leap year.

Date 25/10/2021

Expt. No. 1 Lab-1 Page No. 2

1) Shell Script to check whether a year is a leap year or not

```
#!/bin/sh
echo -n "Enter year: "
read y

if [ $((y %. 4)) -eq 0 ]
then
    if [ $((y %. 400)) -ne 0 -a $
            ((y %. 100)) -eq 0 ]
    then
        echo `Not Leap year'
    else
        echo `Leap year'
    fi
```

Teacher's Signature :

o/p

Enter radius

7

The area is 147

Date 25/10/2021

Expt. No. 2 Page No. 4

2) Shell Script to find Area of Circle

```
#!/bin/sh
echo 'Enter radius'
read r
area=`echo 22/7 \* $r \* $r/bc`
echo 'The area is : $area'
```

Teacher's Signature :

o/p

Enter a number

0

Zero

Expt. No. 3 Date 25/10/2021

Page No. 6

3 Shell Script To check if a number is zero / positive / negative

```
#! bin/sh

echo 'Enter a number'
read n

if [ $n -gt 0 ]
then
    echo 'positive'

elif [ $n -lt 0 ]
then
    echo 'Negative'
else
    echo 'zero'

fi
```

Teacher's Signature :

o/p

```
./big03    28   12   25
28 is the largest
```

Expt. No. 4 Date 25/10/2021

Page No. 8

4) Shell Script to find largest of 3 Nos

```
#!/bin/sh

if [ $# -ne 3]
then
    echo 'Insufficient parameter'
    exit 128
fi

if [ $1 -gt $2 -a $1 -gt $3]
then
    echo "$1 is the largest"

elif [$2 -gt $ -a $2 -gt $3]
then
    echo "$2 is the largest"
else
    echo "$3 is the largest"
```

Teacher's Signature :

o/p:

```
./fac.sh 5
5! = 120
```

Date 8/11/2021

Expt. No. 5 Page No. 19

Lab 2

5 Shell script to compute factorial of a number

```
#!/bin/sh
if [ $# -ne 1 ]
then
    echo 'Insufficient params'
    exit 128
fi
ans=1
n=1
while [ $n -le $1 ]
do
    ans=`expr $ans \* $n`
    n=`expr $n +1`
done
    echo "$1 ! = $ans"
```

Teacher's Signature :

o/p :

./gross - Sal.Sh 1000

Total Salary = 1300

Date 8/11/2021

Expt. No. 6 Page No. 12

6 Compute gross - salary

```
#!/bin/sh
if [ $# -ne 1 ]
then
    echo 'Enter basic salary'
    exit 128
fi

hra = `expr $1 \* 20 /100 | bc`
da = `expr $1 \* 10 /100 | bc`

sal = `expr $1 + $hra + $da | bc`

echo " Total salary = $sal"
```

Teacher's Signature :

o/p

./temp.sh 41

41 degrees in celsius = 5

Date 8-11-2021

Expt. No. 7 Page No. 14

7 Script to Convert temperature from farenheit to celsius

```
#!/bin/sh
if [ $# -ne 1]
then
   echo 'Enter tempr in F'
fi
   x=`expr $1 -32 | bc`
   c=`expr $x \*5 / 9 | bc`
   echo "$1 degrees in celsiy = $c"
```

Teacher's Signature :

o/p

./ arithmetic.sh 20 8

Sum = 28

Difference = 12

Product = 160

Quotient = 2

Remainder = 4

Date 8-11-2021

Expt. No. 8 Page No. 16

8 Script to perform arithmetic on two numbers

```
#!/bin/sh

if [ $# -ne 2]
then
        echo 'enter 2 numbers'
        exit 128
fi
        s=`expr $1 + $2 | bc`
        d=`expr $1 - $2 | bc`
        m=`expr $1 * 2 | bc`
        q=`expr $1 / $2 | bc`
        r=`expr $1 % $2 | bc`

echo "Sum = $s"
echo "Difference = $d"
echo "product = $m"
echo "quotient = $q"
echo "Remainder = $r"
```

Teacher's Signature :

o/p :-

```
bash sum_of_even.sh
6
Sum = 12
```

Date 15/11/2021

Expt. No. 9 Page No. 18

LAB 3

1 Shell Script to find Sum of even numbers upto n

```
#!/bin/bash
if [$# -ne 1]
then
  echo 'Enter range'
  exit 128
fi
Sum=0
for ((i=0; i<=$1; i++))
do
  if ((i%2==0))
  then
    sum=$((i+sum))
  fi
done
echo "Sum = $sum"
```

Teacher's Signature :

o/p :-

```
bash combos.sh

1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1
```

Expt. No. 10 Date 15-11-2021

Page No. 20

10) Print all 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
        if [ $i -ne $j -a $j -ne $k
                -a $i -ne $k ]
        then
          echo "$i -ne $k ]
        fi
        done
    done
```

Teacher's Signature :

o/p:-

bash power.sh 2 4

2 ^ 4 = 16

Date 15-11-2021

Expt. No. 11 Page No. 22

11 Power of a number

```
#! /bin/bash

if [ $# -ne 2 ]
then
  echo 'Enter 2 numbers'
  exit
fi

res = `expr $(($1 ** $2)) bc`

echo "$1 ^ $2 = $res"
```

Teacher's Signature

O/P:

```
bash naturalnosum.sh 3

sum = 6
```

Expt. No. 12 Date 15-11-2021

Page No. 24

12 shell script to find sum of natural numbers

```
#!/bin/bash

if [$# -ne 1]
then
   echo 'Enter range'
   exit 128
fi

  sum=0

for ((i=0; i<=$1; i++))
do
  sum=$((i+sum))
done

echo "sum = $sum"
```

Teacher's Signature :

O/P

Enter CIE marks in all subs

50
48
49
47

Enter SEE in all subs

100
97
95
98

Grade for subject1 is S Grade

Grade for subject2 is S Grade

Grade for subject3 is S Grade

Grade for subject4 is S Grade

Expt. No. 13 Date 29/11/2021

Page No. 26

WEEK - 4

13 Shell Script to display the pass class of a student.

```
#!/bin/bash

# CIE marks

echo "Enter CIE marks in all subs"
cie=()

for i in {0..3}
do
  read cie[$i]
done

#SEE marks

echo "Enter SEE in all subs"
see=()

for i in {0..3}
do
  read see[$i]
done

# calculate marks
```

Teacher's Signature :

Expt. No. 13 Date

Page No. 27

```
totalMarks=()

for i in {0..3}
do
 totalMarks[$i]=`expr ${cie[i]}+${see[i]}/2 | bc`
done

for i in {0..3}
do
 echo "Grade for subject $i is :"

if [ ${totalMarks[$i]} -ge 90 ]
 then
   echo "S Grade"
elfi [ ${totalMarks[$i]} -ge 80 ]
 then
   echo "A Grade"
elfi [ ${totalMarks[$i]} -ge 70 ]
 then
   echo "B Grade"
elfi [ ${totalMarks[$i]} -ge 60 ]
then
   echo "C grade"
elfi echo "E grade"
elfi [ ${totalMarks[$i]} -ge 50 ]
 then
   echo "D grade"
```

Teacher's Signature :

```
elif [ ${totalMarks[$i]} -ge 40]
then
    echo "E grade"
else
    echo "Fail"
fi
done
```

```
elif [ ${totalMarks[$i]} -ge 40 ]
then
    echo "E grade"
else
    echo "Fail"
fi
done
```

Teacher's Signature :

o/p:

```
Enter n: 8
0 1 1 2 3 5 8 13
```

Date 24/11/2021

Expt. No. 14

Page No. 32

14) Shell script to find the Fibonacci series up to n

```
#! /bin/bash

echo -e "Enter n: \c"

read n

first=0
second=1

if ((n >= 1))
then
    printf "%d" $first
fi
if ((n >= 2))
then
    printf "%d" $second
fi

((n=n-2))
while ((n>0))
do
  ((third = first + second))
  printf "%d" $third
  ((first = second))
  ((second = third))
  ((n--))
done
echo # add a newline
```

Teacher's Signature :

O/P

Enter String : Hey there

No. of vowels : 3

Date 29/4/2021

Expt. No. 15 Page No. 34

15 Shell script to count the number of vowels of a string

```
#! /bin/bash

echo -e "Enter a string: \c"
read string

numVowels=`grep -o '[AEIOUaeiou]' <<< $string | wc -l`

echo "No. of Vowels: $numVowels"
```

Teacher's Signature :

o/p :

test.txt

1. Hey there! I'm Vineeth, a CS undergrad.
2. I build websites and software
3. I'm a web developer, software developer and an ethical hacker

fileCounter.sh

bash fileCounter.sh test.txt

Line Count = 2
Word Count = 22
Character Count = 133

Expt. No. 16 Date

Page No. 36

16 Shell Script to check number of lines, words, characters in a file.

```
#!/bin/bash

if [$# -ne 1]
then
    echo 'Enter filename'
    exit 128
fi

echo "Line Count = " `wc -l $1`
echo "Word Count = " `wc -w $1`
echo "Character Count = " `wc -m $1`
```

6/12/2022

Teacher's Signature :

Expt. No. 17 Date 03/01/2022

Page No. 38

Write a C/C++ program that outputs the contents of its Environment list.

```
# include<stdio.h>

int main(int argc, char *argv[]) {
  int i;
  char **ptr;
  extern char ** environ;

  printf("\n\n Start Success\n\n");

  for(ptr = environ; *ptr != 0; ptr++)
   printf("%s\n", *ptr);

  printf("\n\n End Success\n\n");

  return 0;
}
```

Teacher's Signature : _____

Expt. No. 18

Date 3/1/2022

Page No. 39

Write a c/c++ program to emulate the unix ln command.

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

int main(int argc, char *argv[])
{
  if (argc <3 || argc >4 || (argc ==4 &&
                    strcmp(argv [1], "-s"))) {
   printf("usage: ./a.out [-s]<org-file><new_link>\n");
   return 1;
  }

  if (argc ==4) {
   if ((symlink(argv[2], argv[3])) == -1)
    printf("Cannot create symbolic link \n");
   else
    printf("Symbolic link created \n");
  }
  else {
   if ((link(argv[1], argv[2])) == -1)
     printf("cannot create hard link \n");
   else
     printf("Hard link created \n");
   }
   return 0;
  }
```

Teacher's Signature :

Date 3/1/2022

Expt. No. 19 Page No. 40

Write a C/C++ POSIX compliant program that prints the POSIX defined configuration options supported on any given system using feature test macros.

```
#define _posix_source
#define _POSIX_C_SOURCE 199309L
#include &lt;stio.h&gt;

int main()
{
  #ifdef _POSIX_JOB_CONTROL
    printf(&quot; System supports job control \n
                                &quot;);
  #else
    printf(&quot;System does not support job
                control \n &quot;);
  #endif

  #ifdef _POSIX_SAVED_IDS
    printf(&quot; System supports saved set-UID
        and saved set-GID \n &quot;);
  #else
    printf(&quot; System doesnot support saved
        set-UID and saved set-GID \n &quot;);
  #endif
```

Teacher's Signature :

```
#ifdef _POSIX_CHOWN_RESTRICTED
    Printf (&quot;chown-restricted option is %d \n",
    _POSIX_CHOWN_RESTRICTED);
#else
    Printf (&quot;System does not support chown-
            restricted option \n&quot;);
#endif

#ifdef _POSIX_NO_TRUNC
    Printf (&quot;pathname trunc option is %d
            \n&quot;, _POSIX_NO_TRUNC);
#else
    Printf (&quot;System does not support
    system-wide pathname trunc option
            \n&quot);
#endif

#ifdef _POSIX_VDISABLE
    Printf (&quot;Disable character for terminal
    files is %d \n&quot;, _POSIX_VDISABLE);
#else
    printf (&quot;System doesnot support
            _POSIX_VDISABLE \n &quot;);
#endif

    return 0;
}
```

Date 3/1/2022

Expt. No. 20

Page No. 42

Write c/c++ program which demonstrates Interprocess communication between a reader process and a writer process. Use mkfifo, open, read, write and close apis in your program

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

int main(int argc, char *argv[])
{
    int fd;
    char buf[256];

    if (argc != 2 && argc != 3) {
        printf ("USAGE %s <file> [<arg>]\n", argv[0]);

    if (argc == 2) {
        fd = open (argv[1], O_RDONLY | O_NONBLOCK);

        while (read (fd, buf, sizeof (buf)) > 0)
            printf ("%s", buf);
    }
```

Teacher's Signature : _____

```
else {
    fd = open(argv[1], O_WRONLY);
    write(fd, argv[2], strlen(argv[2]));
}

close(fd);
}
```