

Particulars of the Experiments Performed

CONTENTS

Expl No.	Date	Experiment	Marks Obtained	Page No.
1.	25/10/21	Shell script to find leap year & not		1
2.	25/10/21	Shell script to find area of circle.		2
3.	25/10/21	Shell script to check zero/positive /negative		3
4.	25/10/21	Shell script to find biggest of three numbers	full	4
5.	8/11/21	Shell script to find factorial of a number	8/11/21	5
6.	8/11/21	Shell script to find gross salary		6
7.	8/11/21	Shell script to convert fahrenheit to celsius		7
8.	8/11/21	Shell script to perform arithmetic operations		8
9.	15/11/21	Shell script for sum top of even upto n	full 15/11/21	9
10.	15/11/21	Shell script print combination of 123		10
11.	15/11/21	Shell script power of a number		11
12.	15/11/21	Shell script find sum of n natural numbers.		12
13.		Shell script display pass class		13
14.		Shell Script to find Fibonacci series upto n		14
15.		Shell script to count no. of lines, words, chars		15
16.		Shell script count no. of vowels		16
17.		C/C++ prog that output environment variables		17
18.		C/C++ prog to emulate ls command		18
19.		C/C++ Posix comp. prog that prints POSIX defined options.		19
20.		C/C++ Prog which demonstrates Interprocess communication bet. reader, writer procs.		20
				21
				22
				23
				24
				25
				26
				27
				28
				29
				30
				31
				32
				33
				34
				35
				36
				37
				38
				39
				40
				41
				42
				43
				44
				45
				46
				47
				48
				49
				50
				51
				52
				53
				54
				55
				56
				57
				58
				59
				60
				61
				62
				63
				64
				65
				66
				67
				68
				69
				70
				71
				72
				73
				74
				75
				76
				77
				78
				79
				80
				81
				82
				83
				84
				85
				86
				87
				88
				89
				90
				91
				92
				93
				94
				95
				96
				97
				98
				99
				100
				101
				102
				103
				104
				105
				106
				107
				108
				109
				110
				111
				112
				113
				114
				115
				116
				117
				118
				119
				120
				121
				122
				123
				124
				125
				126
				127
				128
				129
				130
				131
				132
				133
				134
				135
				136
				137
				138
				139
				140
				141
				142
				143
				144
				145
				146
				147
				148
				149
				150
				151
				152
				153
				154
				155
				156
				157
				158
				159
				160
				161
				162
				163
				164
				165
				166
				167
				168
				169
				170
				171
				172
				173
				174
				175
				176
				177
				178
				179
				180
				181
				182
				183
				184
				185
				186
				187
				188
				189
				190
				191
				192
				193
				194
				195
				196
				197
				198
				199
				200
				201
				202
				203
				204
				205
				206
				207
				208
				209
				210
				211
				212
				213
				214
				215
				216
				217
				218
				219
				220
				221
				222
				223
				224
				225
				226
				227
				228
				229
				230
				231
				232
				233
				234
				235
				236
				237
				238
				239
				240
				241
				242
				243
				244
				245
				246
				247
				248
				249
				250
				251
				252
				253
				254
				255
				256
				257
				258
				259
				260
				261
				262
				263
				264
				265
				266
				267
				268
				269
				270
				271
				272
				273
				274
				275
				276
				277
				278
				279
				280
				281
				282
				283
				284
				285
				286
				287
				288
				289
				290
				291
				292
				293
				294
				295
				296
				297
				298
				299
				300
				301
				302
				303
				304
				305
				306
				307
				308
				309
				310
				311
				312
				313
				314
				315
				316
				317
				318
				319
				320
				321
				322
				323
				324
				325
				326
				327
				328
				329
				330
				331
				332
				333
				334
				335
				336
				337
				338
				339
				340
				341
				342
				343
				344
				345
				346
				347
				348
				349
				350
				351
				352
				353
				354
				355
				356
				357
				358
				359
				360
				361
				362
				363
				364
				365
				366
				367
				368
				369
				370
				371
				372
				373
				374
				375
				376
				377

Output:

Enter year 1900	not a leap year
Enter year 2000	leap year
Enter year 2021	not a leap year

/

Expt. No. 1

Shell script to find if the given year is leap or not

```
#!/bin/sh
echo "Enter the year"
read year
a=`expr
if [ `expr $year % 100` -eq 0 ]
then
if [ `expr $year % 400` -eq 0 ]
then
echo "leap year"
else
echo "not leap year"
fi
elif [ `expr $year % 4` -eq 0 ]
then
echo "leap year"
else
echo "not leap year"
fi
```

Revised
25/10/21

Teacher's Signature :

Output:

Enter radius of circle 30 2827.3500

Enter radius of circle 7 153.9335

Expt. No. 2

Shell script to find the area of a circle

#!/bin/sh

pi = 3.1415

echo "enter radius of circle"

read r

ans = `echo \$pi * \$r * \$r | bc`

echo \$ans

Teacher's Signature :

Expt. No. 3

Shell Script to check whether the number is zero/positive/negative.

```
#!/bin/sh
echo "enter number"
read num
if [ $num -eq 0 ]
then
echo "zero"
elif [ $num -lt 0 ]
then
echo "negative"
else
echo "positive"
fi
```

N
25/10/21

Teacher's Signature :

Expt. No. 4

Shell script to find largest of 3 numbers

```
#!/bin/sh
echo "Enter 3 numbers"
read a b c
if [ $a -ge $b ]
then
    if [ $a -ge $c ]
    then
        echo "$a is greatest"
    else
        echo "$c is greatest"
    fi
else
    if [ $b -ge $c ]
    then
        echo "$b is greatest"
    else
        echo "$c is greatest"
    fi
fi
```

Teacher's Signature : _____

Output:

Enter 3 numbers -10 -10 0

0 is greatest

Enter 3 numbers 2 2 2

2 is greatest

Enter 3 numbers -2 -3 -1

-1 is greatest

✓

Output:

Enter a number 5

factorial = 120

Enter a number 0

factorial = 1

Enter a number -10

Enter a positive number

Expt. No. 5

Shell script To find factorial of a number

```
#!/bin/bash
echo "Enter a number to find factorial"
read num
fact = 1
forwards
for
if [ $num -lt 0 ]
then
    echo "Enter a positive number"
    exit 2
fi
for ((i=$num; i<=$num; i++ ))
do
    fact=$((fact * i))
done
echo "factorial = $fact"
```

8/11/21

Teacher's Signature : _____

Output:

Enter basic salary 5000 6500

Enter basic salary 1000 1300



Shell Script to convert Fahrenheit to Celsius

#!/bin/sh

echo "enter temperature in fahrenheit"

read ftemp

atemp = `echo \$ftemp -32 | bc`

ctemp = `echo \$atemp * 5 | bc`

echo "scale=2 ; \$ctemp /9;" | bc

✓
N
8/11/21

Output

Enter 2 numbers 59 20

sum = 79,
difference = 39,
multiply = 1180,
divide = 2.95.

Enter 2 numbers 8 3

sum = 11,
difference = 5,
multiply = 24,
divide = 2.66.

Expt. No. 8

Shell Script to give arithmetic operations on given
2 numbers.

#!/bin/sh

echo "Enter 2 numbers"

read a b

sum = `expr \$a + \$b`

diff = `expr \$a - \$b`

mul = `expr \$a * \$b`

div = `echo "\$a / \$b" | bc`

echo "sum = \$sum, difference = \$diff, multiply = \$mul,
divide = \$div"

Teacher's Signature :

Expt. No. 9

Shell script to find sum of even numbers upto n
#!/bin/sh
echo "Enter the value of n"
read n
num = 2
while [num -le \$n]
do
 sum = \$((\$sum + num))
 num = \$((\$num + 2))
done
echo "Sum = \$sum"

✓
N
✓
1/1/21

Teacher's Signature :

Output:

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

.

3 3 3

Expt. No. 10

Date 15/11/21

Page No. 10

Shell script to find combinations of numbers 1 2 3

```
#!/bin/sh
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
15/11/21
```

Teacher's Signature :

Expt. No. 11

Shell script to find power of a number

```

#!/bin/sh
echo "Enter number"
read num
echo "Enter positive power"
read pow
if [ $num -eq 0 ]
then echo "ans = $0"
fi
elif [ $pow -eq 0 ]
then
echo "ans = 1" → elif [ $num -eq 1 ]
else
count = 1
ans = 1
while [ $count -le $pow ]
do
ans = $(($ans * num))
count = $(($count + 1))
done
echo "$ans = $ans"
fi
    
```

Teacher's Signature :

Output:

Enter number 2	$\Rightarrow 16$
Enter power 4	
Enter number 1	$\Rightarrow 1$
Enter power 1000	
Enter number 0	$\Rightarrow 0$
Enter power 200	
Enter number 100	$\Rightarrow 1$
Enter power 0	

Output:

Enter n 3
 Enter number 10
 Enter number 25
 Enter number -1 $\Rightarrow 24$

Enter n 2
 Enter number 0
 Enter number -20 $\Rightarrow -20$

Expt. No. 12

Shell script to find sum of n numbers

```
#!/bin/sh
echo "Enter n"
read n
sum = 0
while [ $n -gt 0 ]
do
  echo "Enter number"
  read num
  sum==$((sum + num))
  n=$((n-1))
done
echo "Sum = $sum"
```

R
Solved

Expt. No. 13

Shell Script to display pass class of a student

```
$ gedit pass.sh
echo "enter marks"
read m
if [ $m -le 100 -a $m -gt 90 ]
then
    echo "grade S"
    echo "PASS"
elif [ $m -le 90 -a $m -gt 80 ]
then
    echo "grade A"
    echo "PASS"
elif [ $m -le 80 -a $m -gt 70 ]
then
    echo "grade B"
    echo "PASS"
elif [ $m -le 70 -a $m -gt 60 ]
then
    echo "grade C"
    echo "PASS"
elif [ $m -le 60 -a $m -ge 50 ]
then
    echo "grade D"
    echo "PASS"
elif [ $m -le 50 ]
then
```

Teacher's Signature : _____

Output

enter marks
100
grade = S
PASS

enter marks
85
grade = A
PASS

enter marks
30
grade = F
FAIL

Date _____

Page No. 14

Expt. No. _____

```
echo "grade F"  
echo "FAIL"  
else  
    echo "enter valid marks"  
fi
```

Teacher's Signature : _____

Shell Script to print fibonacci series upto n

```
$ gedit fib.sh
echo "Enter N"
read N
a=0
b=1
echo "The Fibonacci series is : "
for (( i=0 ; i<N ; i++ ))
do
    echo "$a"
    f=$((a+b))
    a=$b
    b=$f
done
```

Teacher's Signature : _____

OUTPUT:

enter a line of text
i have 6 friends

the given string has 5 vowels, 7 constants and 1 number.

enter a line of text
the fox said he wants 76 fruits.

the given string has 8 vowels, 15 constants and 2 numbers.

Expt. No. 15

Date _____

Page No. 16

Shell script to find the no. of vowels in a string

```
echo "Enter a line of text"
```

```
Read String
```

```
numcount = $(echo $string | grep -o [0-9] | wc --lines)
```

```
vowcount = $(echo $string | grep -o [aeiou] | wc --lines)
```

```
conscount = $(echo $string | grep -o [bcdfghjklmnpqrstvwxyz] | wc --lines)
```

```
echo "the given string has $vowcount vowels, $conscout
```

```
Constants and $numcount numbers in it"
```

Teacher's Signature : _____

Expt. No. 16

Shell Script to check no. of lines, words, characters in a file.

```

echo "enter filename"
read file
c = `cat $file | wc -c`
w = `cat $file | wc -w`
l = `cat $file | grep -c "\."`  

echo "no. of characters in $file is $c"
echo "no. of words in $file is $w"
echo "no. of lines in $file is $l"

```

Teacher's Signature : _____

OUTPUT:

enter a filename:
Vowels.sh

No. of characters in vowels.sh is 320
No. of words in vowels.sh is 49
No. of lines in vowels.sh is 6.

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

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

Expt. No. 18

Write 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 : _____

OUTPUT

Usage : ./a.out [-s] <org-file> <new-link>
[root @ localhost uspi] # ./a.out 1 2 3 4

Usage : ./a.out [-s] <org-file> <new-link>
[root @ localhost uspi] # ./a.out 1.c2

Hard link created

[root @ localhost uspi] # ls -l
-rw-r--r-- 2 root root 657 Mar 27 16:44 1a.c
-rw-r--r-- 2 root root 657 Mar 28 16:44 2 (bolded)
columns are hardlinked count & inode number respectively)

[root @ localhost uspi] # ./a.out 1a.c2
Cannot create hard link (Because 2 already exists)

[root @ localhost uspi] # ./a.out -s 1a.c 22
Symbolic link created

[root @ localhost uspi] # ./a.out -s 1a.c 22
ls -l
-rws-r--r-- root root 657 Mar 27 16:44 1a.c
(rwxrwxrwx l root root 4 Apr 18:32 22 -> 1a.c

[root @ localhost uspi] # readlink 22
1a.c.

OUTPUT

System supports job control
System supports set UID and saved set GID
chown-restricted option is 1
Pathname trunc option is 1
Disable character for terminal files is 0

Expt. No. 19

Date _____

Page No. 20

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_PDSI_X_C_SOURCE 199309L
#include <stdio.h>
#include <unistd.h>
int main()

#ifdef _POSIX_JOB_CONTROL
printf("System supports job control\n");
#endif
#ifdef _POSIX_SAVED_IDS
printf("System supports set UID and saved set GID\n");
#else
printf("System doesn't support IDs");
#endif
#ifdef _POSIX_CHOWN_RESTRICTED
printf("chown-restricted option is %d\n", _POSIX_CHOWN_RESTRICTED);
#endif
printf("System doesn't support chown-restricted option\n");
#endif
#ifndef _POSIX_NO_TRUNC
printf("Pathname trunc option is %d\n", _POSIX_NO_TRUNC);
#endif.
```

Teacher's Signature : _____

Date _____

Expt. No. _____

Page No. 21

```
#ifdef _POSIX_VDISABLE
printf ("Disable character for terminal file is %d \n", _POSIX_VDISABLE);
#else
printf ("System doesn't support _POSIX_DISABLE \n");
#endif
return 0;
}
```

OUTPUT:

```
/* Terminal 1- writer process */
[root @localhost ~] # ./a.out FIFO1 "This is USP + CD lab"
After this open new terminal by pressing shift + Ctrl + N & go
to file → open terminal
```

```
/* Terminal 2- reader process */
[root @localhost ~] # ./a.out FIFO1
```

This is USP and CDlab.

Expt. No. 20

Date _____
Page No. 22

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

```
#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 != 2 && argc != 3)
    {
        printf ("USAGE ./s <fd> [<args>] \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);
    }
}
```

Teacher's Signature : _____

Date _____

Expt. No. _____

Page No. 23

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