UNIX LAB MANUAL
SUBJECT CODE: BCA 404P
Class IV sem BCA
19-Batch

Positional parameter

UNIX AND SHELL PROGRAMMING LAB

PART – A home1

1. Write a shell script to print all the prime numbers between m to n (m<n).
2. Write a shell script to reverse a given string and check whether it is a palindrome.
3. Write a shell script to find the sum of digits of a given number using loops and without loops.
4. Write a shell script to implement 10 unix commands using case .
5. Write a Shell script that displays list of all the files in the current directory to which the user has read, write and execute permissions?
6. Write a shell script to copy a file
i) copy file within current directory. ii)copy file between two directories.
7. Write a Shell script to create 2 data files and compare them to display unique and common entries.
8. Write awk script to find number of words, characters and lines in a file.
9. Write a system program to demonstrate fork(),exec(),wait(),exit() system calls to execute ls –l command in the child process.
10. Write a Menu driven program to demonstrate zombie process and orphan process.

PART-B

11. Write a shell script to find the factorial of a given number.
12. Write a shell script to count the number of vowels in a given string using case.
13. Create a file containing the following fields: Student No, student name, age, gender, height and weight. Print all the details in a neat format.
14. Write shell script to find area and circumference of a circle.
15. Write a shell script to print first "n" even and odd numbers.
16. Write a shell program to demonstrate grep command.
17. Shell script to compress and decompress a file using gzip and gunzip commands.
clear
18. Write a Shell script to find given pattern using command line arguments
19. To check whether the file exists or not if it exists give details of its attributes like access permission, size ,time, date etc
20. Write shell script to show pattern matching using meta-characters.

Instructions to execute shell programs

Start -> run -> type telnet 10.0.0.29

Username: bca

Password : bca123

$ vi prime.sh -> to create / open shell program

Type the program and then press esc shift : wq → it will save and quit

$ sh prime.sh -> to execute shell program

Program 1 : Write shell script to print all prime numbers between m and n(m<n).

```
echo "enter 2 numbers to find prime between 2 nos"
read  low
read   high
echo "prime numbers between $low and $high is"
while test  $low   -lt   $high
do
flag=0
lo=`expr   $low   /   2`
for((i=2;i <= lo;i++))
do
if  test  `expr $low   %  $i`   -eq   0
then
flag=1
break
fi
done #for end

if  test  $flag   -eq    0
then
echo "$low"
fi
low=`expr   $low + 1`
done  #while end
```

```
echo "enter 2 numbers"
read low
read high

echo "prime numbers between $low and $high is"

while [ $low -lt $high ]
do
  flag=0
  lo=`expr $low / 2`

  for((i=2;i <= lo; i++))
  do
       if [ `expr $low % $i` -eq 0 ]
       then
           flag=1
       break
       fi

  done #for end

  if [ $flag -eq 0 ]
  then
          echo "$low"
  fi
  low=`expr $low + 1`
done  #while end
```

Program2: Write a shell program to reverse a given number and check whether it is palindrome or not.

#wc is counting total characters in str

cut command will cut line by characters column wise

rev1 is empty initially. Then concating letter by letter

```
echo enter the string
read str
len=`echo $str | wc -c`
len=`expr $len - 1`
rev1=""
while test $len -ge 1
do
                rev=`echo $str | cut -c $len`
                rev1=$rev1$rev
                len=`expr $len - 1`
done

                echo reverse of $str=$rev1
                if test $str = $rev1
                        then
                                echo it is palindrome
                        else
                                echo it is not palindrome
                        fi
```

Program 3: Write a shell program to find sum of digits of a given number.

```
sum=0
function using_loop()
{
        while [ $n -ne 0 ]
        do
                sum=$((sum+n%10))
                n=$((n/10))
        done
}
function using_recursive()
{
        if [ $n -ne 0 ]
        then
                sum=$((sum+n%10))
                n=$((n/10))
                using_recursive
        fi
}
echo "Enter a number"
read n
echo "How do you want to calculate the sum of digits?"
echo "1: using loop"
echo "2: using recursive function"
read ch
case $ch in
        1) using_loop ;;
        2) using_recursive ;;
esac
```

```
echo
echo -n "Sum of digits = "
echo $sum

root@CA-ANITHA:~# cd programs
root@CA-ANITHA:~/programs# vi digit.sh
root@CA-ANITHA:~/programs# bash digit.sh
Enter a number
123
How do you want to calculate the sum of digits?
1: using loop
2: using recursive function
1

Sum of digits = 6
root@CA-ANITHA:~/programs# bash digit.sh
Enter a number
376
How do you want to calculate the sum of digits?
1: using loop
2: using recursive function
2

Sum of digits = 16
root@CA-ANITHA:~/programs#
```

#program 4 - Write a shell script to implement 10 unix commands using case .

the command ;; indicates that the program flow should jump to the end of the entire case statement. This is similar to break in the C programming language.

PROGRAM4

```
ans="y"
while [  $ans  =  "y"  ]
do
tput clear
echo "MENU OPTION"
tput cup 7 5
echo "1. Date command....."
tput cup 9 5
echo "2. pwd command......"
tput cup 11 5
echo "3. calendar command."
tput cup 13 5
echo "4. who command......"
tput cup 15 5
echo "5. Terminal Type...."
```

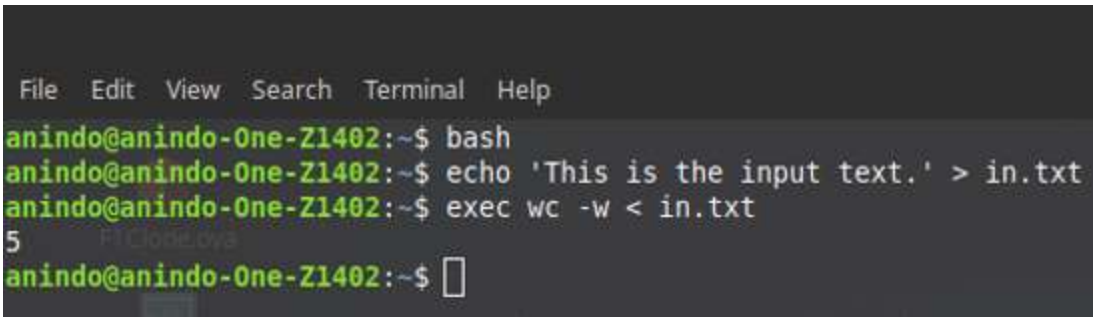
```
tput cup 17 5
echo "6. cp command....."
tput cup 19 5
echo "7. cat command....."
tput cup 21 5
echo "8. wc command....."
tput cup 23 5
echo "9. calculator command"
tput cup 25 5
echo "10. ls command....."
tput cup 27 5
echo "Enter your choice"
tput cup 27 25
read ch
case $ch in
        1)date;;
        2)pwd;;
        3)cal;;
        4)who;;
        5)tty;;
        6)echo "Enter the source filename"
          read f1
          echo "Enter the destination filename"
          read f2
          cp $f1 $f2;;
        7)echo "Enter a filename"
          read  fname
          cat  $fname;;
        8)echo "Enter a filename"
          read fname
          wc $fname;;
        9)bc;;
           10)ls -l;;
        *)echo "Invalid choice"
esac
echo
echo
echo "Do you want to continue?"
read ans
done
```

PROGRAM 5

Write a Shell script that displays list of all the files in the current directory to which the user has read, write and execute permissions?

The **read** command **reads** the file **line** by **line**, assigning each **line** to the $**line** bash shell variable. Once all **lines** are **read** from the file the bash while loop will stop.

exec command in Linux is used to execute a command from the bash itself. This command does not create a new process it just replaces the bash with the command to be executed. If the exec command is successful, it does not return to the calling process.




```
echo "enter the directory name"
read dir
if  test  -d $dir
then
   cd $dir
   ls > test
   exec < test
while read line
do
   if  test -f   $line
  then
      if  test  -r   $line   -a   -w   $line   -a   -x   $line
     then
            echo "$line has all permissions"
         else
             echo "$line not having all permissions"
     fi
  fi
done
```

program 6. Write a shell script to copy a file
i) copy file within current directory. ii)copy file between two directories.

```
ans="y"
while [“$ans” = "y"]
do
clear
echo "MENU OPTION"
echo "1. Copy a content of a file to another file"
echo "2. Copy a file from one directory to another"
echo "Enter your choice:"
read ch
case  $ch in
      1)echo "Enter the source file"
        read fname1
        echo "Enter the destination file"
        read fname2
        cp $fname1 $fname2
        echo "Content of source file"
        cat $fname1
        echo "Content of destination file"
        cat $fname2;
      2)echo "Enter the source file"
        read fname
        echo "Enter the destination directory"
        read dir
        if   [  ! -d $dir
        then
             echo "Directory does not exist"
        else
            cp $fname $dir;;
        fi
        esac
echo
echo
echo "Do you want to continue?"
read ans
done
```

Program7: Shell script to create 2 data files and compare them to display unique and common entries.

```
Shell script to create 2 data files and compare them to display unique
and common entries. pr7.sh
clear
opt="y"

echo "enter the data for f1.txt"
while test  $opt  == "y"
do

        read data1
        echo $data1>>f1.txt
        echo "do you want to add data (y/n)"
        read opt
done

opt="y"
echo "enter the data for f2.txt"
while test  $opt == "y"
do
                read data2
                echo $data2>>f2.txt
                echo "do you want to add data (y/n)"
                read opt
done

        clear
        while test 1
```

```
do
        echo "MENU DRIVEN"
        echo "1.compare command"
        echo "2.uniq command"
        echo "3.common command"
        echo "4.exit"
        echo "enter your choice"
        read ch
        case $ch in
                1)echo "enter the file names"
                        read fname1
                        read fname2
                        cmp $fname1 $fname2;;
                2)echo "enter the file name"
                        read fname
                        sort $fname | uniq;;
                3)echo "enter the file names"
                        read fname1
                        read fname2
                        sort -o f1.txt $fname1
                        sort -o f2.txt $fname2
                        comm  -1 -2 $fname1 $fname2;;
                4)exit;;
                        *)echo "Invalid choice";;
        esac
done
```

8. Write awk script to find number of words, characters and lines in a file.

Program name - count.awk

```
BEGIN{print "record\t characters \t words"}
#BODY section
{
len=length($0)
total_len =len
print(NR,":\t",len,":\t",NF,$0)
words =NF
}
END{
  print("\n total")
  print("characters :\t" total len)
  print("lines :\t" NR)
```

}

commands

AWK: The awk is similar to sed it can be used for to edit the file.

NF: The NF can be used to count the number of fields on a records.

NR: The NR can be used to count the number of records on a file.

Mv:The mv command is used to move the data from one file to another file.

Syntax: mv file1 file2.

Steps to execute

Create file called "test.txt"
vi test.txt

hello world
How are you

esc shift :wq

```
a.out          count.awk  pr4.sh  prime.sh      zombi.c
bca            f          pr5.sh  program4.sh
commands.sh    fork1.c    pr6.sh  test.txt
[anitha1@localhost ~]$ awk -f count.awk test.txt
record    characters      words
1 :       11 :     2 hello world
2 :       11 :     3 how are you

 total
characters :     11
lines : 2
[anitha1@localhost ~]$
```

9. Write a system program to demonstrate fork(),exec(),wait(),exit() system calls to execute ls –l command in the child process.

```
#include<stdio.h>
main()
{
```

```
        int pid,ch;
        pid=fork();
        if (pid==0)
        {
                printf("Child process id=%d\n",pid);
                printf("Parent process id=%d\n",getppid());
                execl("/bin/ls","ls","-l",NULL);     this is letter L
                printf("This will not be displayed at all");
}
else
{
        wait();
        while(1)
        {
                printf("\nTo exit press 1:\n");
                scanf("%d",&ch);
                if(ch==1)
                exit(0);
                else
                continue;
        }
}
```

****

10. Write a Menu driven program to demonstrate zombie process and orphan process.

```
// C program to execute zombie and
// orphan process in a single program
#include <stdio.h>
int main()
{

   int x;
   x = fork();

   if (x > 0)
      printf("IN PARENT PROCESS\nMY PROCESS ID: %d\n", getpid());

   else if (x == 0) {
      sleep(5);
      x = fork();
```

```
        if (x > 0) {
   printf("IN CHILD PROCESS\nMY PROCESS ID :%d\n PARENT PROCESS ID : %d\n",
getpid(), getppid());
 while(1)
    sleep(1);

   printf("IN CHILD PROCESS\nMY PARENT PROCESS ID
                          : %d\n", getppid());
        }

        else if (x == 0)
           printf("IN CHILD'S CHILD PROCESS\n
                MY PARENT ID : %d\n", getppid());
    }

    return 0;
}
```

PART-B

11. Write a shell script to find the factorial of a given number

$vi fact.sh
Press I ←insert mode

```
clear
echo enter the number
read num
fact=1
while   test  $num -ge 1
do
fact=`expr   $fact    \*   $num`
num=`expr   $num   -   1`
done
echo "factorial =$fact"
```

esc shift : wq

$sh fact.sh

$enter the number

4

Factorial=4

12. Write a shell program to count the number of vowels in a given string using case.

```
# shell program to count the number of vowels in a given

clear
echo enter the string
read str
len=`echo $str | wc -c`
len=`expr $len - 1`
echo $len
count=0
while test $len -ge 1
do
        char=`echo $str | cut -c $len`
        len=`expr $len - 1`
case $char in 'a'|'e'|'i'|'o'|'u'|'A'|'E'|'I'|'O'|'U')
        count=`expr $count + 1`;;
        esac
done
echo total number of vowel=$count
```

13. Create a file containing the following fields: Student No, student name, age, gender, height and weight. Print all the details in a neat format.

```
clear
opt=y
while test $opt = y
do
  echo "enter student number"
  read num
  echo "enter student name"
  read name
  echo "enter student age"
  read age
  echo "enter student gender"
  read s
  echo "enter student weight"
  read wt
  echo "enter student height"
  read ht
  echo "$num|$name|$age|$s|$wt|$ht" >> student
  echo "wish to continue?"
  read opt
done
clear
```

```
echo "student details"
echo "..............."
echo "no    name     age    gender    weight    height"
echo
echo "........................................"
awk -F "|" '{printf "%4d %-20s %4d %-8s %8d %3d \n", $1,$2,$3,$4,$5,$6}' student
echo "........................................"
```

14. Write shell script to find area and circumference of a circle.

```
#WRITE SHELL SCRIPT TO FIND THE AREA AND CIRCUMFERENCE OF A CIRCLE
echo -n "Enter the radius of a circle : "
read r

# use formula to get it
area=`echo "scale=2;3.14 * ($r * $r)" | bc`
circumference=`echo "scale=2; 2 * $r * 3.14 "| bc`

echo "Area of circle is $area"
echo "Circumference of circle is $circumference"
```

15. Write a shell program to print first "n" even and odd numbers.

```
#program 15
#write a shell program to print first N evn and odd Number
echo "Enter the Number "
read n
for((i=0;i<$n;i++))
do
        if [ `expr $i % 2` = 0 ]
        then
                echo " even = $i"
        fi
done
for((i=0;i<$n;i++))
do
                if [ `expr $i % 2` != 0 ]
                then
                        echo "odd = $i"
                fi
done
```

16. Write a shell script to demonstrate grep command.

```
$ cat > myfile.txt
unix is great os. unix is opensource. unix is free os.
learn operating system.
Unix linux which one you choose?
```

```
uNix is easy to learn.unix is a multiuser os.Learn unix .unix is a
powerful.
```

```
#PROGRAM16
#WRITE A SHELL SCRIPT TO DEMONSTRATE GREP COMMAND

echo demonstrate grep command
echo "-------------------------"
echo "enter the file name"
read fname

echo "CASE SENSITIVE SEARCH"
grep -i "UNix" $fname
echo "COUNTING NUMBER OF MATCHES"
grep -c "unix" $fname
echo "DISPLAY FILE NAME THAT MATCHES PATTERN"
grep -l "unix" *
echo "DISPLAY ONLY MATCHED PATTERN"
grep -o "unix" $fname
```

17. Shell script to compress and decompress a file using gzip and gunzip commands.

```
clear
echo 1.gzip
echo 2.gunzip
echo 3.exit
echo enter your choice:
tput cup 3 18
read ch
case $ch in
1)echo enter file to be compressed
read file
gzip $file
echo the file has been compressed
ls -l;;
2)echo enter the file to be uncompressed
read file
gunzip $file
echo the file has been uncompressed
ls -l;;
3)exit
esac
```

18. Write a Shell script to find given pattern using command line arguments

```
#SHELL SCRIPT TO FIND GIVEN PATTERN USING
#COMMAND LINE ARGUMENTS
clear
echo "Program Name $0"
echo "number of arguments $#"
echo "List of arguments $*"
echo "Lines with pattern $1 in $2"
grep $1 $2
#end
```

```
[Radika.CA-ANITHA] ➤ cat > myfile
welcome to unix programming
thank you
welcome

[2018-02-06 21:54.30]
[Radika.CA-ANITHA] ➤ sh pos.sh welcome myfile
Program Name pos.sh
number of arguments 2
List of arguments welcome myfile
Lines with pattern welcome in myfile
welcome to unix programming
welcome
```

program 19

To check whether the file exists or not if it exists give details of its attributes like access permission, size ,time, date etc

```
clear
echo enter the file name
read fname
if [ -e $fname ]
then
echo "$fname exists and its attributes are"
ls -l $fname
else
echo "$fname does not exist"
fi
```

20. Write a shell script to show pattern matching using meta-characters.

```
clear
```

```
echo "Enter a single character please"
read char

case $char in

[A-Z]) echo “You entered a Capital letter” ;;
[a-z]) echo “You entered a lowercase letter” ;;
[0-9]) echo “You entered a digit” ;;
?) echo “Your entered a special symbol” ;;
*) echo “you entered more than one character “ ;;
esac
```