UNP Lab

Lab Exercise -1

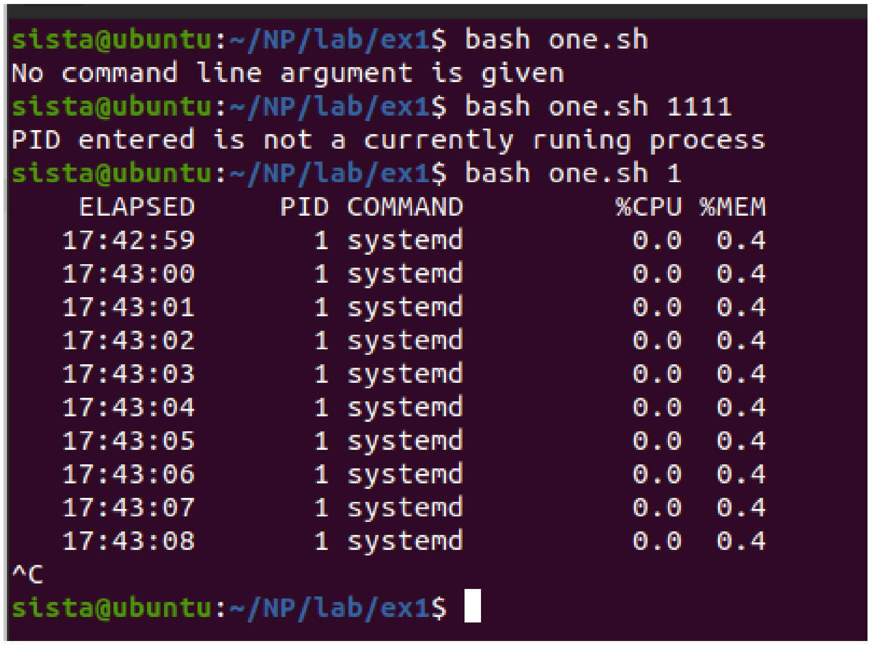
Name:Shravani SIsta

Roll No.:19MCME01

1) one.sh

```
#!/bin/bash
input_pid="$1"
if [ ! -z $input_pid ]
then
	lines=$(ps -eo pid | awk '$1 =='$input_pid'' || wc -l)
	if [ ! -z $lines ]
	then
		ps -eo etime,pid,comm,pcpu,pmem | head -1
		while true
		do
			ps -eo etime,pid,comm,pcpu,pmem | awk '{if($2
=='$input_pid')print $0}'
			sleep 1
		done
	else
		echo "PID entered is not a currently runing process"
	fi
else
	echo "No command line argument is given"
fi
```

Output:



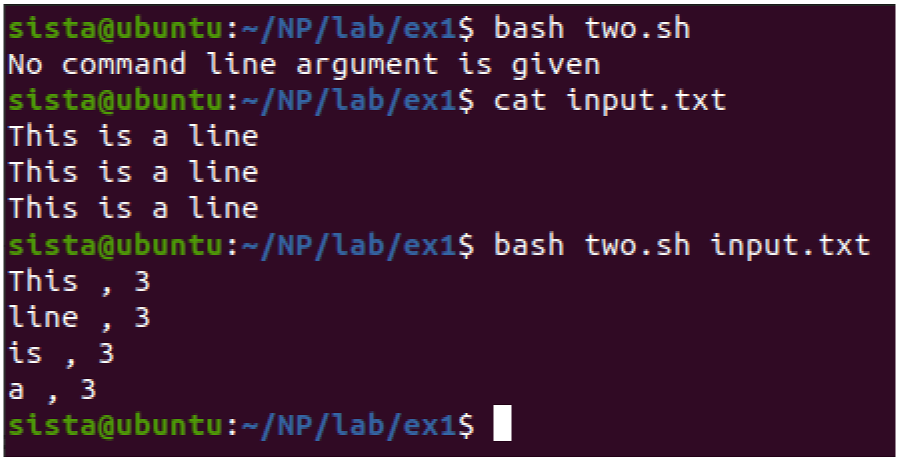
```
sista@ubuntu:~/NP/lab/ex1$ bash one.sh
No command line argument is given
sista@ubuntu:~/NP/lab/ex1$ bash one.sh 1111
PID entered is not a currently runing process
sista@ubuntu:~/NP/lab/ex1$ bash one.sh 1
    ELAPSED     PID COMMAND         %CPU %MEM
   17:42:59       1 systemd          0.0  0.4
   17:43:00       1 systemd          0.0  0.4
   17:43:01       1 systemd          0.0  0.4
   17:43:02       1 systemd          0.0  0.4
   17:43:03       1 systemd          0.0  0.4
   17:43:04       1 systemd          0.0  0.4
   17:43:05       1 systemd          0.0  0.4
   17:43:06       1 systemd          0.0  0.4
   17:43:07       1 systemd          0.0  0.4
   17:43:08       1 systemd          0.0  0.4
^C
sista@ubuntu:~/NP/lab/ex1$
```

2) two.sh

```
#!/bin/bash
file="$1"
if [ ! -z $file ]
then
    declare -A word_count
    while read -r line
    do
      for word in $line
      do
            if [ -v word_count[$word] ]
            then
                 ((word_count[$word]++))
            else
                 word_count[$word]=1
            fi
      done
    done < $file

    for word in "${!word_count[@]}"
    do
      echo $word "," ${word_count[$word]}
    done
else
    echo "No command line argument is given"
fi
```

Output:



```
sista@ubuntu:~/NP/lab/ex1$ bash two.sh
No command line argument is given
sista@ubuntu:~/NP/lab/ex1$ cat input.txt
This is a line
This is a line
This is a line
sista@ubuntu:~/NP/lab/ex1$ bash two.sh input.txt
This , 3
line , 3
is , 3
a , 3
sista@ubuntu:~/NP/lab/ex1$
```

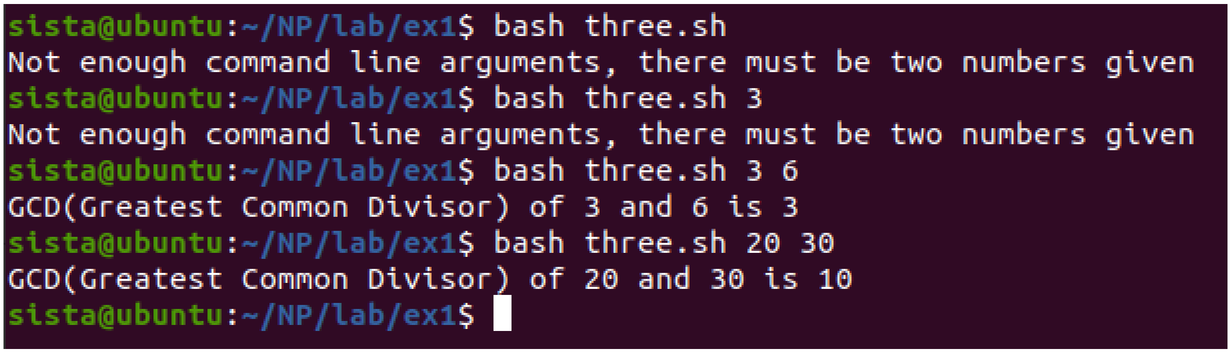
3) three.sh

```
#!/bin/bash
cal_GCD(){
    if [ $2 -ne 0 ]
    then
      temp=`cal_GCD $2 $(($1%$2))`
      echo $temp
    else
      echo $1
    fi
}

num1=$1
num2=$2

if [ ! -z $num1 ] && [ ! -z $num2 ]
then
    GCD=`cal_GCD $num1 $num2`
    echo "GCD(Greatest Common Divisor) of $num1 and $num2 is
$GCD"
else
    echo "Not enough command line arguments, there must be two
numbers given"
fi
```

Output:



```
sista@ubuntu:~/NP/lab/ex1$ bash three.sh
Not enough command line arguments, there must be two numbers given
sista@ubuntu:~/NP/lab/ex1$ bash three.sh 3
Not enough command line arguments, there must be two numbers given
sista@ubuntu:~/NP/lab/ex1$ bash three.sh 3 6
GCD(Greatest Common Divisor) of 3 and 6 is 3
sista@ubuntu:~/NP/lab/ex1$ bash three.sh 20 30
GCD(Greatest Common Divisor) of 20 and 30 is 10
sista@ubuntu:~/NP/lab/ex1$
```

4) a.sh

```
#!/bin/bash
echo "Executing a.sh"
ps
exec "./b.sh"
```

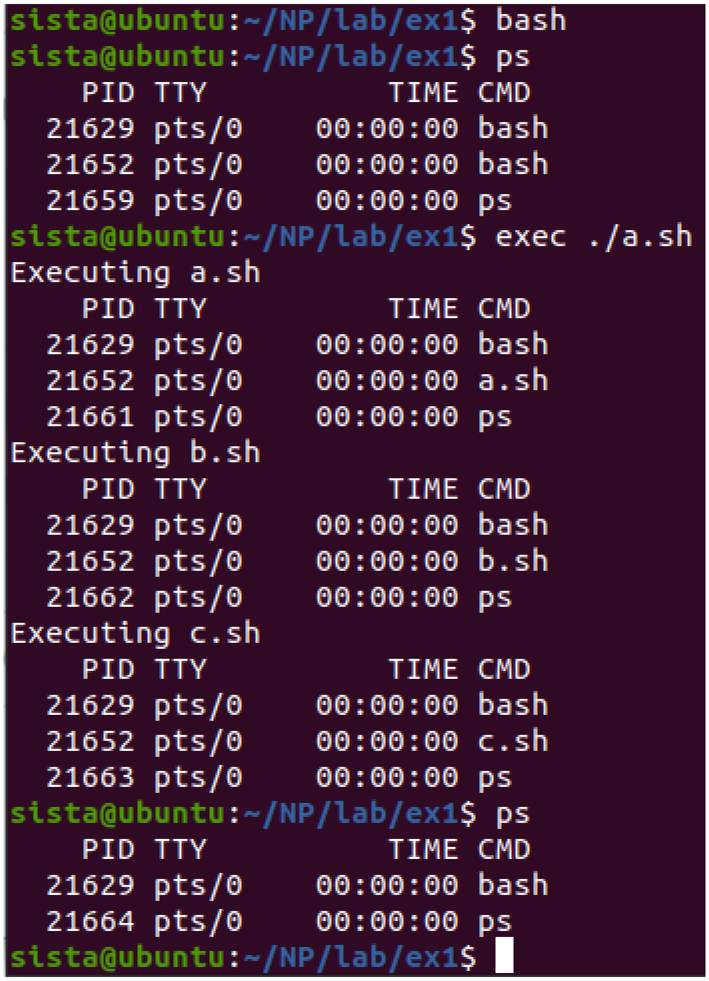
b.sh

```
#!/bin/bash
echo "Executing b.sh"
ps
exec "./c.sh"
```

c.sh

```
#!/bin/bash
echo "Executing c.sh"
ps
```

Output:



```
sista@ubuntu:~/NP/lab/ex1$ bash
sista@ubuntu:~/NP/lab/ex1$ ps
    PID TTY          TIME CMD
  21629 pts/0    00:00:00 bash
  21652 pts/0    00:00:00 bash
  21659 pts/0    00:00:00 ps
sista@ubuntu:~/NP/lab/ex1$ exec ./a.sh
Executing a.sh
    PID TTY          TIME CMD
  21629 pts/0    00:00:00 bash
  21652 pts/0    00:00:00 a.sh
  21661 pts/0    00:00:00 ps
Executing b.sh
    PID TTY          TIME CMD
  21629 pts/0    00:00:00 bash
  21652 pts/0    00:00:00 b.sh
  21662 pts/0    00:00:00 ps
Executing c.sh
    PID TTY          TIME CMD
  21629 pts/0    00:00:00 bash
  21652 pts/0    00:00:00 c.sh
  21663 pts/0    00:00:00 ps
sista@ubuntu:~/NP/lab/ex1$ ps
    PID TTY          TIME CMD
  21629 pts/0    00:00:00 bash
  21664 pts/0    00:00:00 ps
sista@ubuntu:~/NP/lab/ex1$ 
```

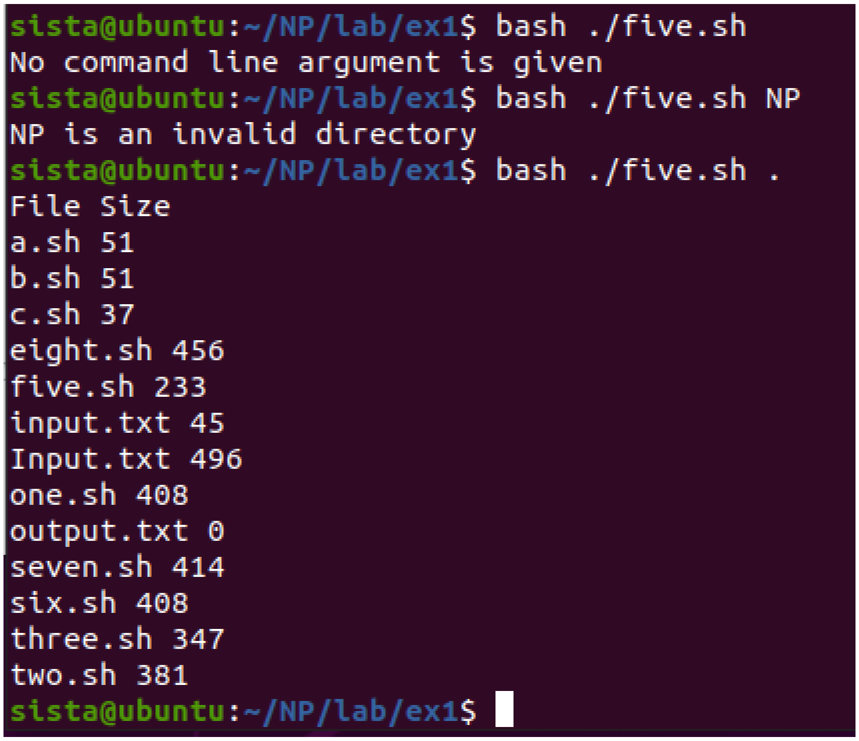
Observations:
All files when executed share the same PID. What exec does is it replaces the current process image with a new process image. The fact that all files share the same PID when executed demonstrates this.

5) five.sh

```
#!/bin/bash
dir="$1"
if [ ! -z $dir ]
then
    if [ -d $dir ]
    then
```

```
      echo "File Size"
      ls -l $dir | tail -n+2  | awk '{print $9, $5}'
    else
      echo "$dir is an invalid directory"
    fi
else
    echo "No command line argument is given"
fi
```

Output:



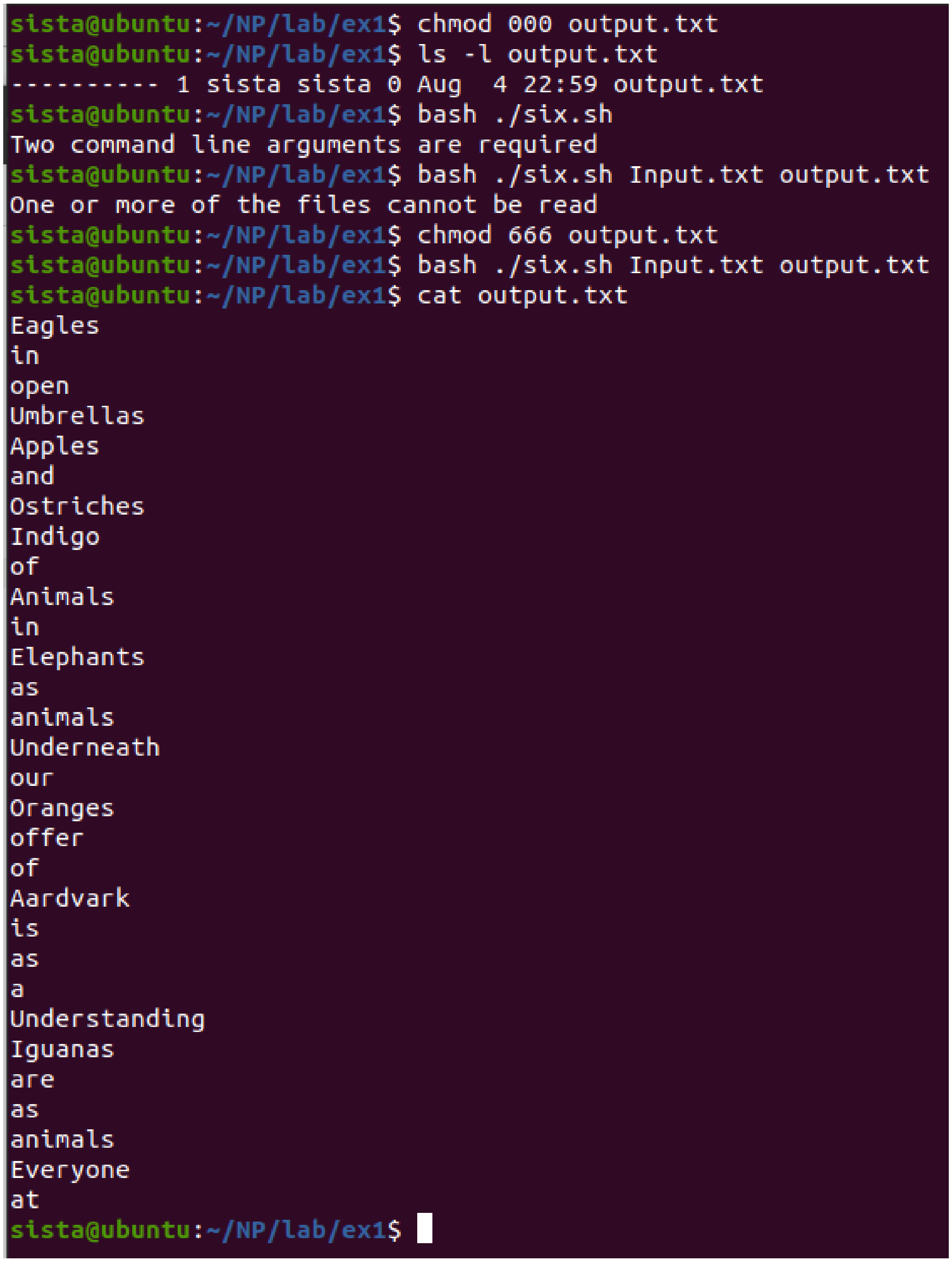
```
sista@ubuntu:~/NP/lab/ex1$ bash ./five.sh
No command line argument is given
sista@ubuntu:~/NP/lab/ex1$ bash ./five.sh NP
NP is an invalid directory
sista@ubuntu:~/NP/lab/ex1$ bash ./five.sh .
File Size
a.sh 51
b.sh 51
c.sh 37
eight.sh 456
five.sh 233
input.txt 45
Input.txt 496
one.sh 408
output.txt 0
seven.sh 414
six.sh 408
three.sh 347
two.sh 381
sista@ubuntu:~/NP/lab/ex1$ 
```

6) six.sh

```
#!/bin/bash
input_file="$1"
output_file="$2"
if [ ! -z $input_file ] && [ ! -z $output_file ]
then
    if [ -r $input_file ] && [ -w $output_file ]
    then
      while read -r line
      do
            for word in $line
            do
                 echo $word | grep -x "[AEIOUaeiou].*" >>
$output_file
```

```
            done
      done < $input_file
    else
      echo "One or more of the files cannot be read"
    fi
else
    echo "Two command line arguments are required"
fi
```

Output:

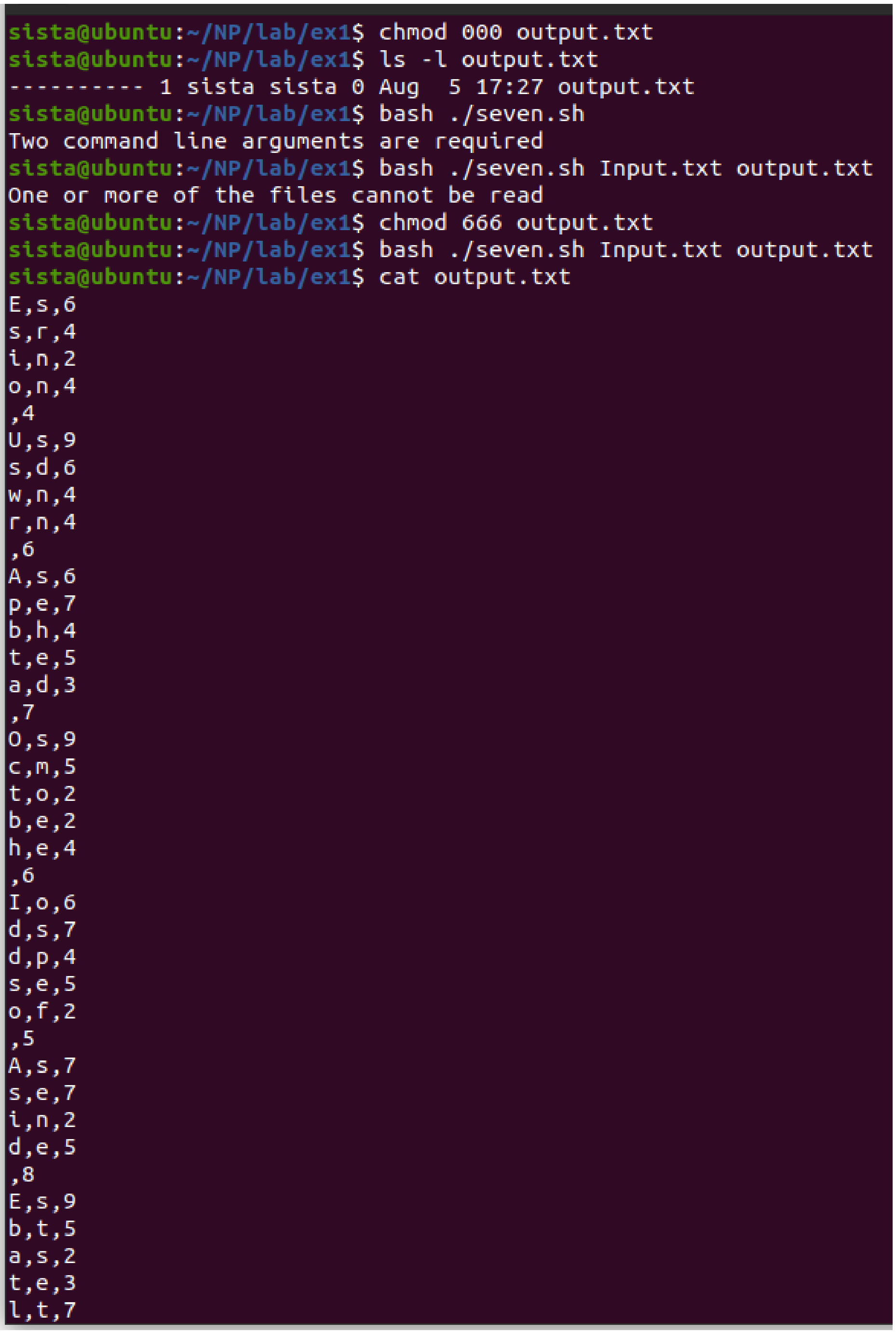


```
sista@ubuntu:~/NP/lab/ex1$ chmod 000 output.txt
sista@ubuntu:~/NP/lab/ex1$ ls -l output.txt
---------- 1 sista sista 0 Aug  4 22:59 output.txt
sista@ubuntu:~/NP/lab/ex1$ bash ./six.sh
Two command line arguments are required
sista@ubuntu:~/NP/lab/ex1$ bash ./six.sh Input.txt output.txt
One or more of the files cannot be read
sista@ubuntu:~/NP/lab/ex1$ chmod 666 output.txt
sista@ubuntu:~/NP/lab/ex1$ bash ./six.sh Input.txt output.txt
sista@ubuntu:~/NP/lab/ex1$ cat output.txt
Eagles
in
open
Umbrellas
Apples
and
Ostriches
Indigo
of
Animals
in
Elephants
as
animals
Underneath
our
Oranges
offer
of
Aardvark
is
as
a
Understanding
Iguanas
are
as
animals
Everyone
at
sista@ubuntu:~/NP/lab/ex1$ 
```

7) seven.sh

```
#!/bin/bash
input_file="$1"
output_file="$2"
if [ ! -z $input_file ] && [ ! -z $output_file ]
then
    if [ -r $input_file ] && [ -w $output_file ]
    then
      while read -r line
      do
            for word in $line
            do
                 echo ${word:0:1}","${word: -1}","${#word}  >>
$output_file
            done
      done < $input_file
    else
      echo "One or more of the files cannot be read"
    fi
else
    echo "Two command line arguments are required"
fi
```

Output:



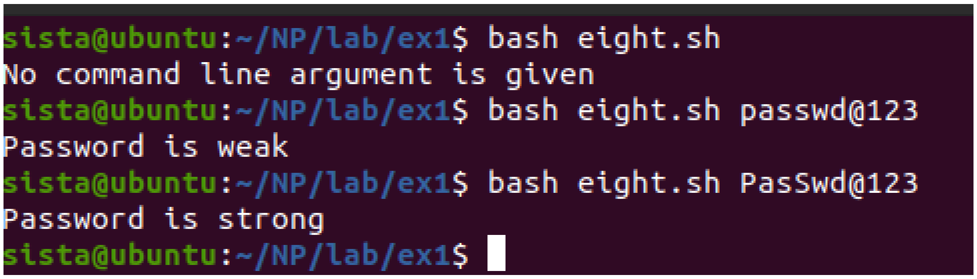
```
sista@ubuntu:~/NP/lab/ex1$ chmod 000 output.txt
sista@ubuntu:~/NP/lab/ex1$ ls -l output.txt
---------- 1 sista sista 0 Aug  5 17:27 output.txt
sista@ubuntu:~/NP/lab/ex1$ bash ./seven.sh
Two command line arguments are required
sista@ubuntu:~/NP/lab/ex1$ bash ./seven.sh Input.txt output.txt
One or more of the files cannot be read
sista@ubuntu:~/NP/lab/ex1$ chmod 666 output.txt
sista@ubuntu:~/NP/lab/ex1$ bash ./seven.sh Input.txt output.txt
sista@ubuntu:~/NP/lab/ex1$ cat output.txt
E,s,6
s,r,4
i,n,2
o,n,4
,4
U,s,9
s,d,6
w,n,4
r,n,4
,6
A,s,6
p,e,7
b,h,4
t,e,5
a,d,3
,7
O,s,9
c,m,5
t,o,2
b,e,2
h,e,4
,6
I,o,6
d,s,7
d,p,4
s,e,5
o,f,2
,5
A,s,7
s,e,7
i,n,2
d,e,5
,8
E,s,9
b,t,5
a,s,2
t,e,3
l,t,7
```

8) eight.sh

```
#!/bin/bash
pass="$1"
if [ ! -z $pass ]
then
    upper_count=$(echo $pass | grep -o [A-Z] | tr -d "\n"] | wc
-m)
    lower_count=$(echo $pass | grep -o [a-z] | tr -d "\n"] | wc
-m)
    if [ ${#pass} > 9 ] && [[ $pass =~ [[:alpha:]] && $pass =~
[[:digit:]] ]] && [ $upper_count -ge 2 ] && [ $lower_count -ge
2 ] && [[ $pass =~ ^[A-Z](.*)$ ]]
    then
      echo "Password is strong"
    else
      echo "Password is weak"
    fi

else
    echo "No command line argument is given"
fi
```

Output:



```
sista@ubuntu:~/NP/lab/ex1$ bash eight.sh
No command line argument is given
sista@ubuntu:~/NP/lab/ex1$ bash eight.sh passwd@123
Password is weak
sista@ubuntu:~/NP/lab/ex1$ bash eight.sh PasSwd@123
Password is strong
sista@ubuntu:~/NP/lab/ex1$ 
```