

Experiment 2

1. Obtain the system time, and check whether it is in the morning, afternoon, or evening.

```
#!/bin/bash
hour=`date +%H`
case $hour in
0[1-9]|1[01])
    echo "Good morning !!"
    ;;
1[2-7])
    echo "Good afternoon !!"
    ;;
*)
    echo "Good evening !!"
    ;;
esac
```

```
yjz@yjz:~/B22170220$ vim demo1.sh
yjz@yjz:~/B22170220$ chmod u+x demo1.sh
yjz@yjz:~/B22170220$ ./demo1.sh
Good morning !!
```

2. Input two number, check which one is greater, and output the result.

```
#!/bin/bash
echo "Enter the first integer:"
read first
echo "Enter the second integer:"
read second
if [ "$first" -gt "$second" ]
then
    echo "$first is greater than $second"
elif [ "$first" -lt "$second" ]
then
    echo "$first is less than $second"
else
    echo "$first is equal to $second"
fi
```

```
yjz@yjz:~/B22170220$ vim demo2.sh
yjz@yjz:~/B22170220$ chmod u+x demo2.sh
yjz@yjz:~/B22170220$ ./demo2.sh
```

```
yjz@yjz:~/B22170220$ ./demo2.sh
Enter the first integer:
11
Enter the second integer:
18
11 is less than 18
yjz@yjz:~/B22170220$ ./demo2.sh
Enter the first integer:
11
Enter the second integer:
11
11 is equal to 11
yjz@yjz:~/B22170220$ ./demo2.sh
Enter the first integer:
11
Enter the second integer:
9
11 is greater than 9
```

3. Find the minimal value in a given list.

```
#!/bin/bash
smallest=10000

for i in 8 2 18 0 -3 87
do
    if test $i -lt $smallest
    then
        smallest=$i
    fi
done
echo "The smallest number is: $smallest"
```

```
yjz@yjz:~/B22170220$ vim demo3.sh
yjz@yjz:~/B22170220$ chmod u+x demo3.sh
yjz@yjz:~/B22170220$ ./demo3.sh
The smallest number is: -3
yjz@yjz:~/B22170220$ 
```

4. Calculate the number of executable file in the current directory.

```
#!/bin/bash
count=0
for i in *
do
    if test -x $i
    then
        count=`expr $count+1`
    fi
done
echo Total of $count executable files
```

The screenshot shows a terminal window titled 'ubuntu' in Xshell 8. The session is connected via SSH to the IP address 192.168.17.134:22. The terminal displays the following command-line interaction:

```
yjz@yjz:~/B22170220$ ls
demo1.sh demo2.sh demo3.sh
yjz@yjz:~/B22170220$ vim demo4.sh
yjz@yjz:~/B22170220$ chmod u+x demo4.sh
yjz@yjz:~/B22170220$ ./demo4.sh
Total of 0+1+1+1+1 executable files
yjz@yjz:~/B22170220$ vim demo4.sh
yjz@yjz:~/B22170220$ ./demo4.sh
Total of 0+1+1+1+1 executable files
yjz@yjz:~/B22170220$
```

The terminal window includes standard Linux navigation keys (arrow keys, home, end) and a status bar at the bottom showing 'SSH2', 'xterm', '77x39', '10,22', '1 会话', and various keyboard function indicators like CAP, NUM, and arrow keys.

5.Check whether a given number is a prime, you have to write a function, and call the function.

```
#!/bin/bash

prime( )
{
    flag=1
    j=2
    while [ $j -le `expr $1 / 2` ]
    do
        if [ `expr $1 % $j` -eq 0 ]
        then
            flag=0
            break
        fi
        j=`expr $j + 1`
    done
    if [ $flag -eq 1 ]
    then
        return 1
    else
        return 0
    fi
}

prime $1

if [ $? -eq 1 ]
then
    echo "$1 is a prime!"
else
    echo "$1 is not a prime!"
fi
```

```
yjz@yjz:~/B22170220$ vim demo5.sh
yjz@yjz:~/B22170220$ ./demo5.sh 1
1 is a prime!
yjz@yjz:~/B22170220$ ./demo5.sh 12
12 is not a prime!
yjz@yjz:~/B22170220$ ./demo5.sh 13
13 is a prime!
yjz@yjz:~/B22170220$ █
```