Experiment 2

use a editor to finishe the following shell scripts, and run them in Linux system.

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 morining !!"
 ;;
1[234567] )
 echo "Good afternoon !!"
 ;;
 * )
 echo "Good evening !! "
 ;;
Esac
```

```
wch@wch-VirtualBox:~$ chmod u+x 2-1.sh
wch@wch-VirtualBox:~$ ./2-1.sh
Good morining !!
wch@wch-VirtualBox:~$ date +%H
11
wch@wch-VirtualBox:~$
```

打开(O) 2-1.sh ~/

```
1  #!/bin/bash
2  hour=`date +%H`
3   case $hour in
4   0[1-9] | 1[01] )
5   echo "Good morining !!"
6   ;;
7  1[234567] )
8   echo "Good afternoon !!"
9   ;;
10  * )
11  echo "Good evening !! "
12  ;;
13 esac
```

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

```
#!/bin/sh
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
```

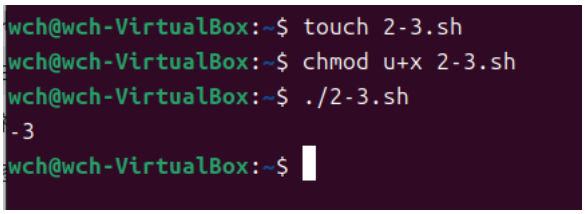
```
wch@wch-VirtualBox:~$ chmod u+x 2-2.sh
wch@wch-VirtualBox:~$ ./2-2.sh
Enter the first integer:
10
Enter the second integer:
20
10 is less than 20
wch@wch-VirtualBox:~$ ./2-2.sh
Enter the first integer:
10
Enter the second integer:
10
10 is equal to 10
wch@wch-VirtualBox:~$ ./2-2.sh
Enter the first integer:
20
Enter the second integer:
10
20 is greater than 10
wch@wch-VirtualBox:~$
```

打开(O)　2-2.sh ~/

```
1 #!/bin/sh
2 echo "Enter the first integer:"
3 read first
4 echo "Enter the second integer:"
5 read second
6 if [ "$first" -gt "$second" ]
7  then
8  echo "$first is greater than $second
9  elif [ "$first" -lt "$second" ]
10 then
11  echo "$first is less than $second"
12 else
13  echo "$first is equal to $second"
14 fi
```

3. Find the minimal value in a given list.

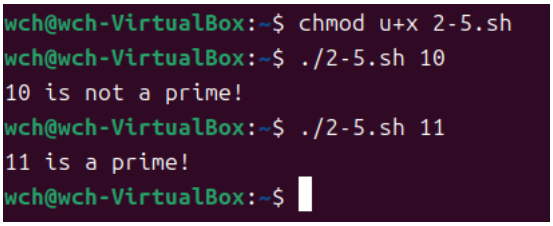
```
#!/bin/bash
smallest=10000 for i in 8 2 18 0 -3 87
do
if test $i -lt $smallest
then
smallest=$i
fi
done
echo $smallest
```



4. Calculate the number of executive 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 files executable
```

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



打开(O)　　2-5.sh ~/

```
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
```