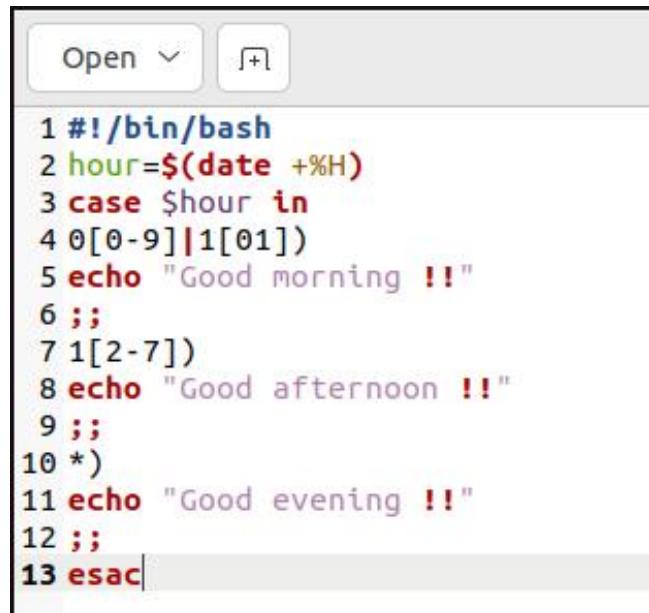


# Experiment 2

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
```



A screenshot of a terminal window showing the script code. The window has a title bar with 'Open' and a '+' icon. The code is displayed in red and blue syntax highlighting:

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

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo b23040920 zouxin
b23040920 zouxin
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ touch script1.sh script2.sh script3.sh script4.sh script5.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script1.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script1.sh
bash: ./script1.sh: Permission denied
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ chmod +x script1.sh script2.sh script3.sh script4.sh script5.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script1.sh
./script1.sh: line 2: hour: command not found
./script1.sh: line 13: syntax error near unexpected token `newline'
./script1.sh: line 13: `Esa'
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ edit script1.sh
Error: no "edit" mailcap rules found for type "application/x-sh"
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script1.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script1.sh
Good morning !!
```

## 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"
Ft
```

```
Open ▾ +
```

```
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
```

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo b23040920 zouxin
b23040920 zouxin
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script2.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script2.sh
Enter the first integer:
3
Enter the second integer:
6
3 is less than 6
```

### 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
```

```
Open ▾ +
```

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

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo b23040920 zouxin
b23040920 zouxin
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script3.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script3.sh
-3
```

#### 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
```

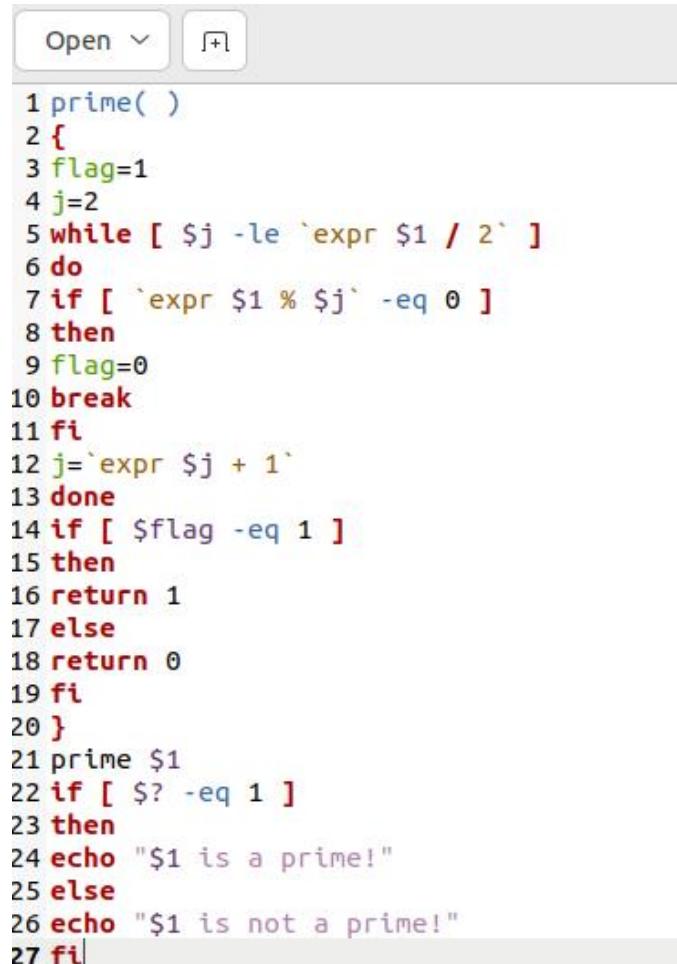
```
Open ▾ +  
1 #!/bin/bash  
2 count=0  
3 for i in *  
4 do  
5 if test -x $i  
6 then  
7 count=`expr $count + 1`  
8 fi  
9 done  
10 echo Total of $count files executable
```

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo b23040920 zouxin  
b23040920 zouxin  
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script4.sh  
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script4.sh  
Total of 5 files executable
```

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

```
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
```



```
1 prime( )
2 {
3     flag=1
4     j=2
5     while [ $j -le `expr $1 / 2` ]
6     do
7         if [ `expr $1 % $j` -eq 0 ]
8         then
9             flag=0
10        break
11    fi
12    j=`expr $j + 1`
13 done
14 if [ $flag -eq 1 ]
15 then
16     return 1
17 else
18     return 0
19 fi
20 }
21 prime $1
22 if [ $? -eq 1 ]
23 then
24     echo "$1 is a prime!"
25 else
26     echo "$1 is not a prime!"
27 fi
```

```
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ echo b23040920 zouxin
b23040920 zouxin
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ gedit script5.sh
zouxin@zouxin-virtual-machine:~/Desktop/B23040920liunx$ ./script5.sh
expr: syntax error: unexpected argument '2'
./script5.sh: line 5: [: 2: unary operator expected
is a prime!
```