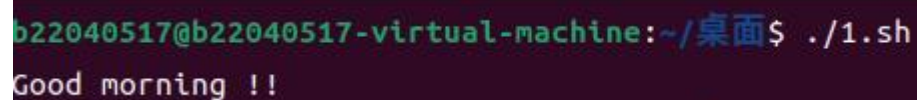


1. Obtain the system time, and check whether it is in the morning, afternoon, or evening.

```
#!/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
```



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



```
b22040517@b22040517-virtual-machine:~/桌面$ ./1.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"
fi
```



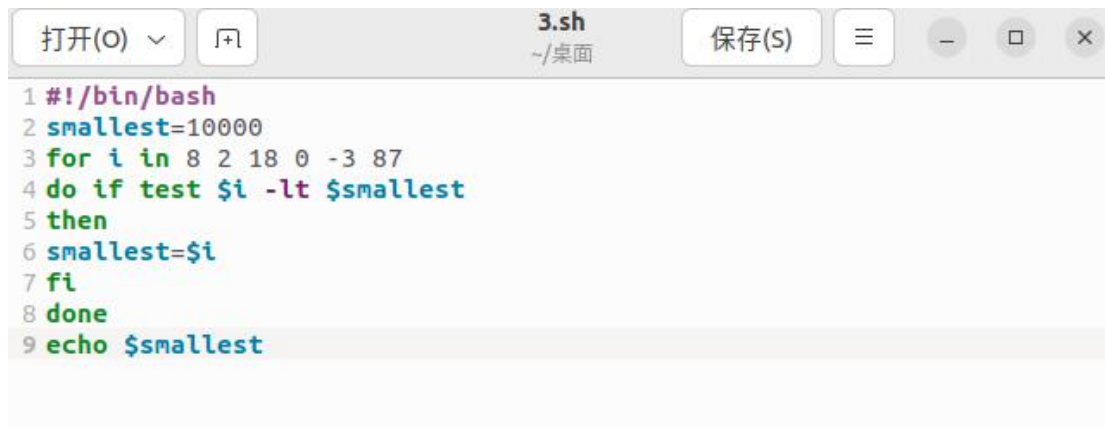
A screenshot of a Zsh terminal window. The title bar shows 'Z.sh' and the path '~/桌面'. The window contains the same shell script as shown in the previous block, with line numbers 1 through 14 on the left margin. The script prompts for two integers and compares them using -gt, -lt, and -eq operators.

```
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the first integer:
3
Enter the second integer:
4
3 is less than 4
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the first integer:
2
Enter the second integer:
2
2 is equal to 2
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the first integer:
4
Enter the second integer:
1
4 is greater than 1
```

3. Find the minimal value in a given list.

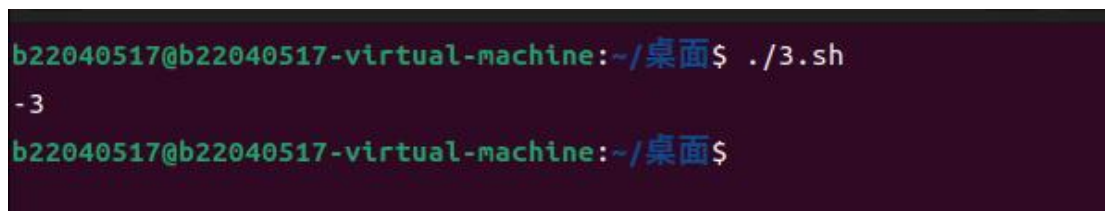
```
#!/bin/bash  
smallest=10000
```

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



A screenshot of a text editor window titled "3.sh" with a path of "~/桌面". The window contains the following script:

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



A screenshot of a terminal window showing the execution of the script. The prompt is "b22040517@b22040517-virtual-machine:~/桌面\$". The command ". /3.sh" is entered, and the output is "-3".

```
b22040517@b22040517-virtual-machine:~/桌面$ ./3.sh  
-3  
b22040517@b22040517-virtual-machine:~/桌面$
```

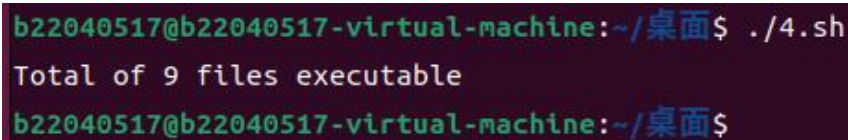
4. Calculate the number of executive file in the current directory.

```
#!/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
```



The screenshot shows a code editor window with a title bar containing '4.sh' and the file path '~/桌面'. The editor has standard buttons for '打开(O)', '保存(S)', and window controls. The script content is displayed with syntax highlighting: line numbers 1-10, keywords in blue, variables in green, and strings in red. Line 7 is highlighted with a light blue background.

```
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
```



The screenshot shows a terminal window with a dark background. The prompt is 'b22040517@b22040517-virtual-machine:~/桌面\$'. The user enters './4.sh', and the output is 'Total of 9 files executable'. The prompt then changes to 'b22040517@b22040517-virtual-machine:~/桌面\$'.

```
b22040517@b22040517-virtual-machine:~/桌面$ ./4.sh
Total of 9 files executable
b22040517@b22040517-virtual-machine:~/桌面$
```

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
```

```
5.sh
~/桌面
保存(S)

1 #!/bin/bash
2
3 prime() {
4     local num=$1
5     local flag=1
6     local j=2
7     while [ $j -le $((num / 2)) ]
8     do
9         if [ $((num % j)) -eq 0 ]
10        then
11            flag=0
12            break
13        fi
14        j=$((j + 1))
15    done
16    echo $flag
17 }
18
19 if [ -z "$1" ]; then
20     echo "Please provide a number to check:"
21     read input
22 else
23     input=$1
24 fi
25
26 result=$(prime $input)
27 if [ $result -eq 1 ]
28 then
29     echo "$input is a prime!"
30 else
31     echo "$input is not a prime!"
32 fi
```

```
b22040517@b22040517-virtual-machine:~/桌面$ ./5.sh
Please provide a number to check:
4
4 is not a prime!
b22040517@b22040517-virtual-machine:~/桌面$ ./5.sh
Please provide a number to check:
3
3 is a prime!
b22040517@b22040517-virtual-machine:~/桌面$
```