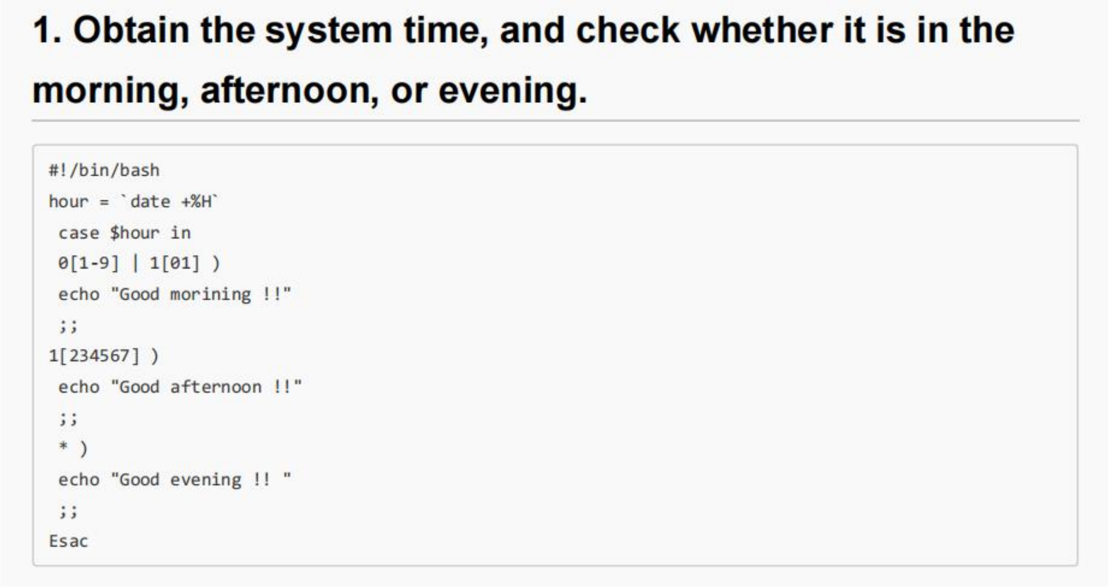
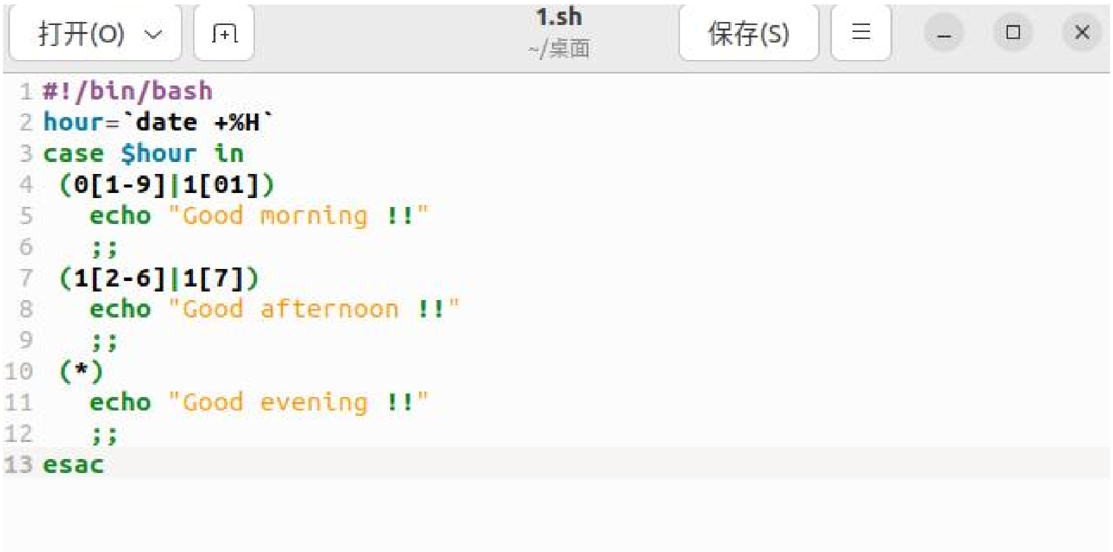
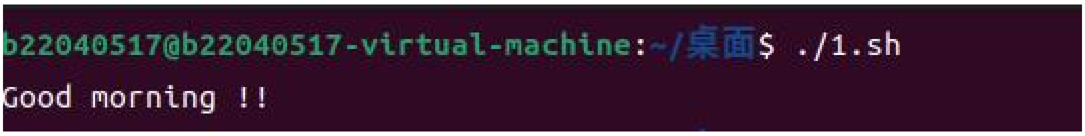


1. Obtain the system time, and check whether it is in the morning, afternoon, or evening.

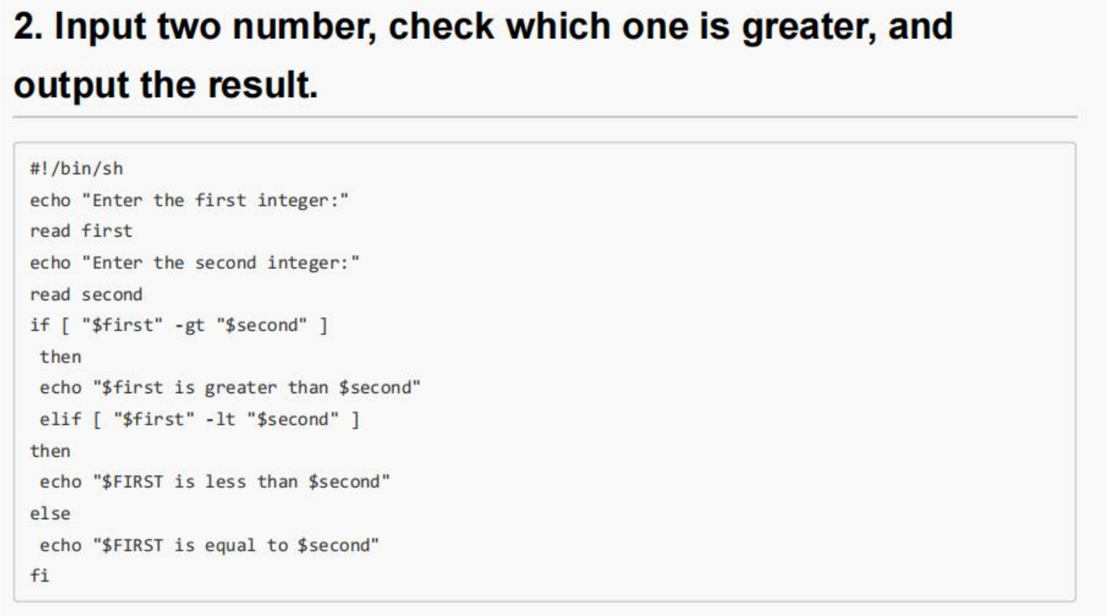
```
#!/bin/bash
hour = `date +%H`
 case $hour in
 0[1-9] | 1[01] )
 echo "Good morining !!"
 ;;
1[234567] )
 echo "Good afternoon !!"
 ;;
 * )
 echo "Good evening !! "
 ;;
Esac
```



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

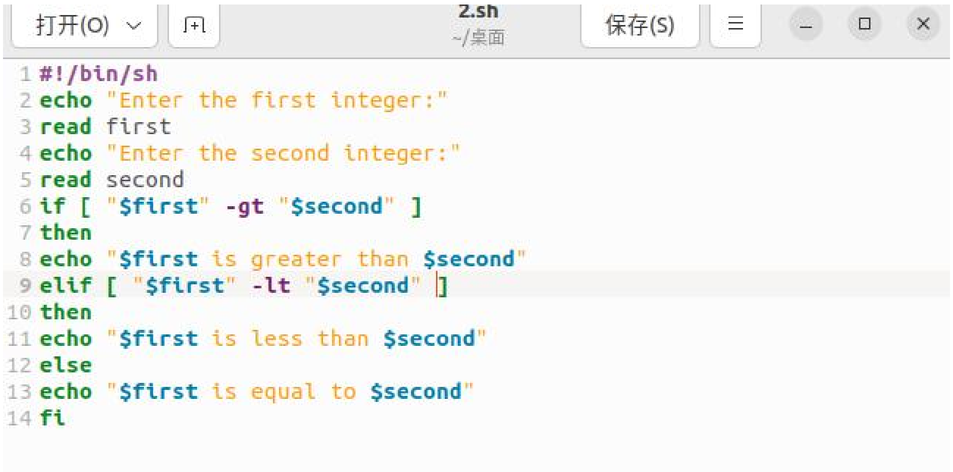


```
b22040517@b22040517-virtual-machine:~/桌面$ ./1.sh
Good morning !!
```

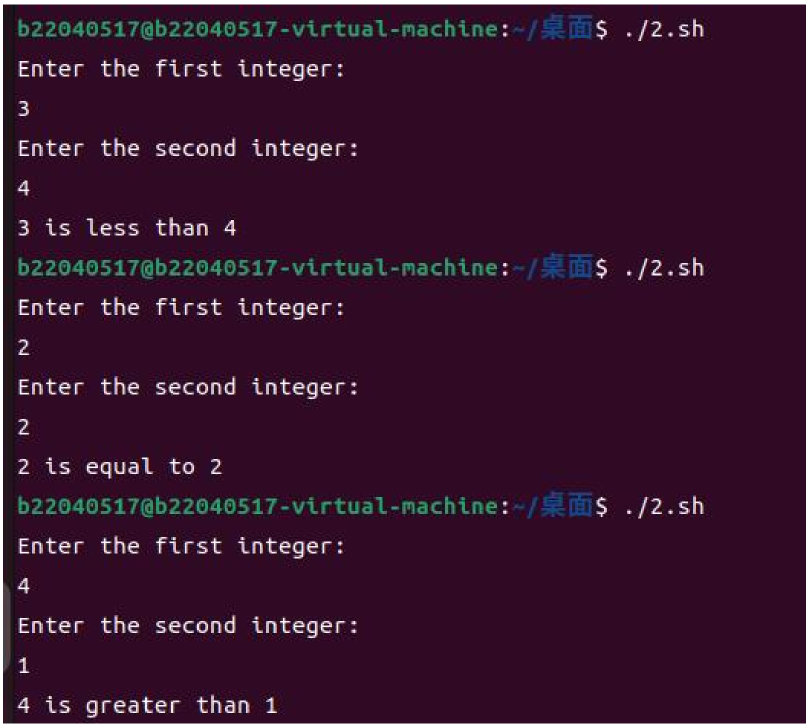


2. Input two number, check which one is greater, and output the result.

```
#!/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
```



```
#!/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
```

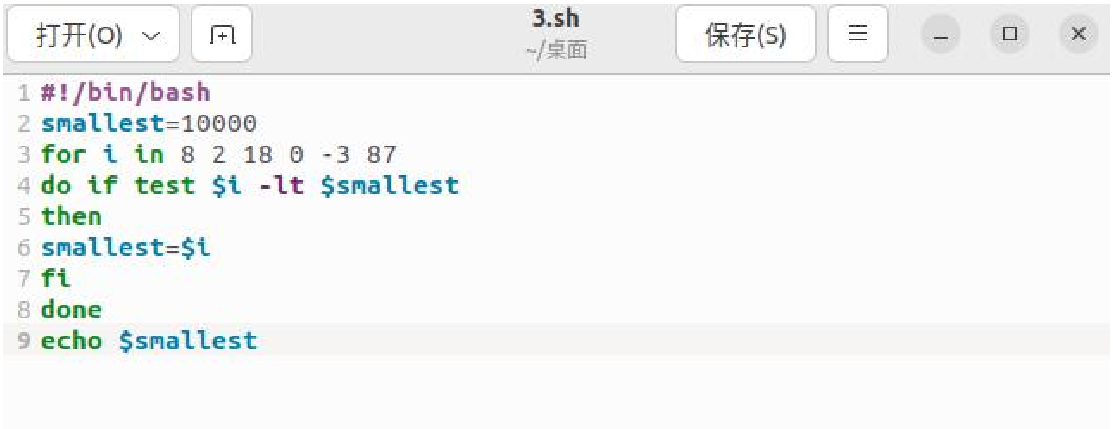


```
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the first integer:
3
Enter the second integer:
4
3 is less than 4
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the first integer:
2
Enter the second integer:
2
2 is equal to 2
b22040517@b22040517-virtual-machine:~/桌面$ ./2.sh
Enter the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

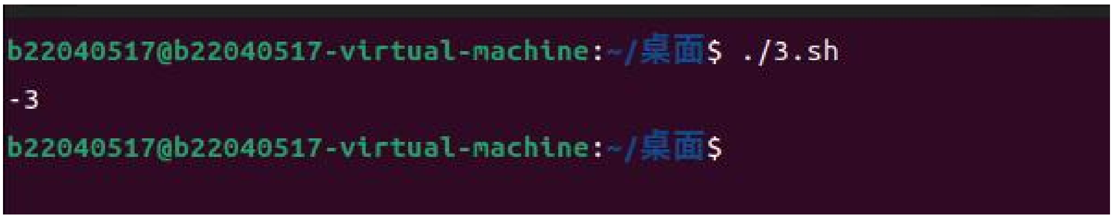
3. Find the minimal value in a given list.

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

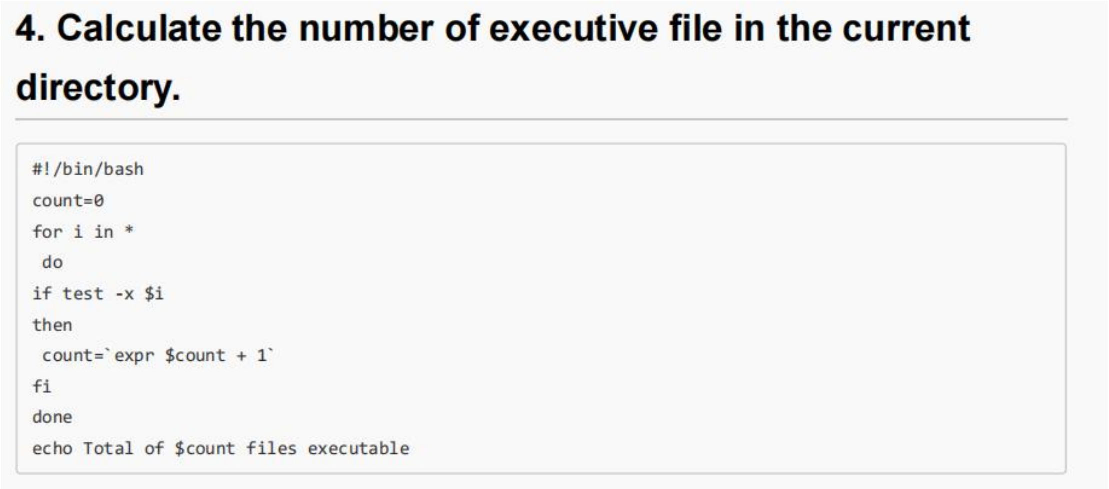
```
for i in 8 2 18 0 -3 87
do
if test $i -lt $smallest
then
 smallest=$i
fi
done
echo $smallest
```



```
#!/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
```

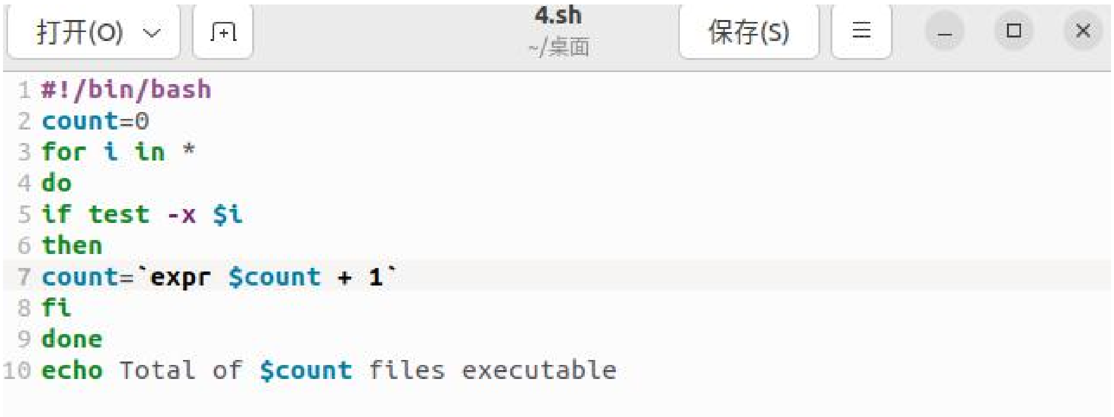


```
b22040517@b22040517-virtual-machine:~/桌面$ ./3.sh
-3
b22040517@b22040517-virtual-machine:~/桌面$
```

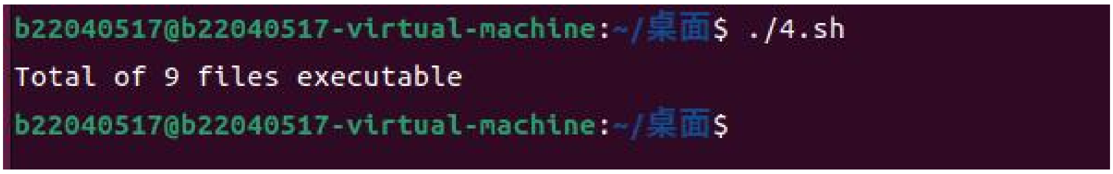


4. Calculate the number of executive file in the current directory.

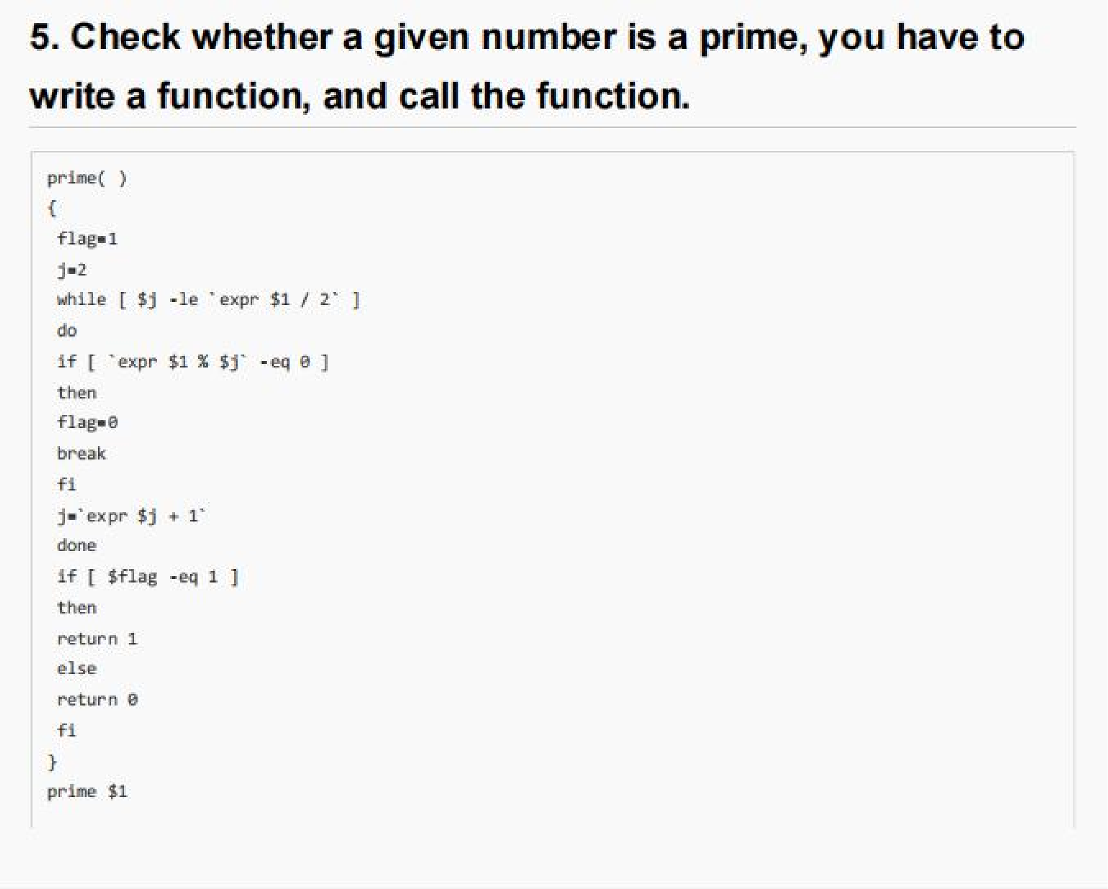
```
#!/bin/bash
count=0
for i in *
 do
if test -x $i
then
 count=`expr $count + 1`
fi
done
echo Total of $count files executable
```



```
#!/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
```

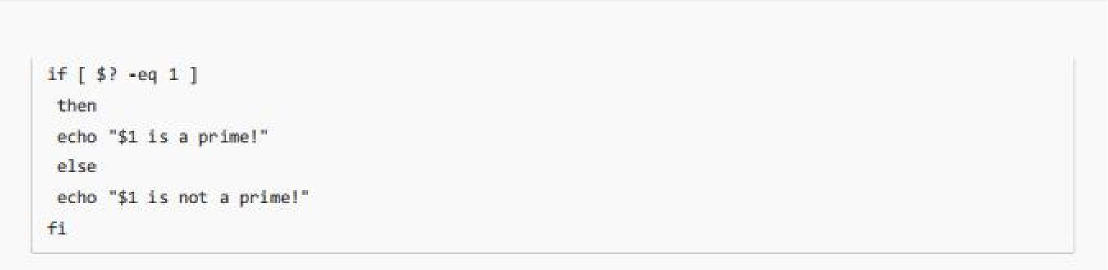


```
b22040517@b22040517-virtual-machine:~/桌面$ ./4.sh
Total of 9 files executable
b22040517@b22040517-virtual-machine:~/桌面$
```

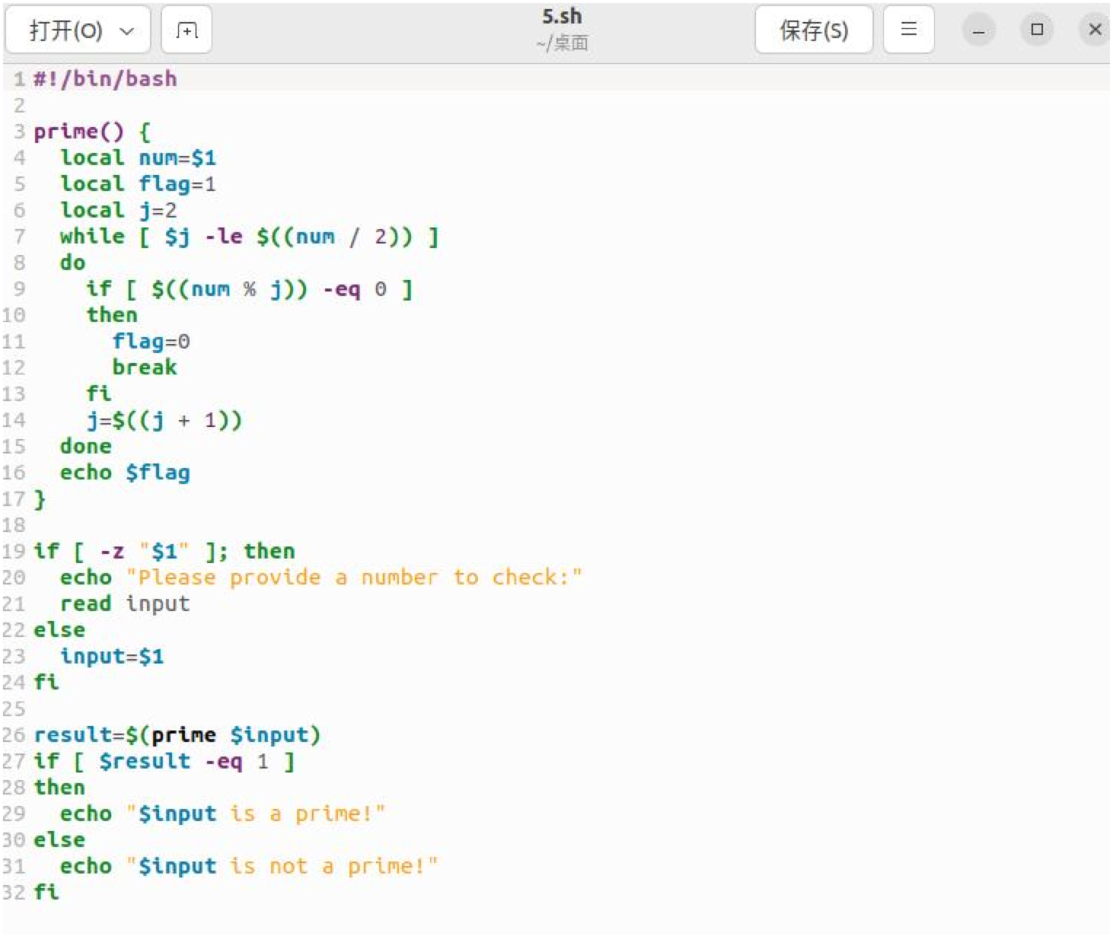


5. Check whether a given number is a prime, you have to write a function, and call the function.

```
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
```

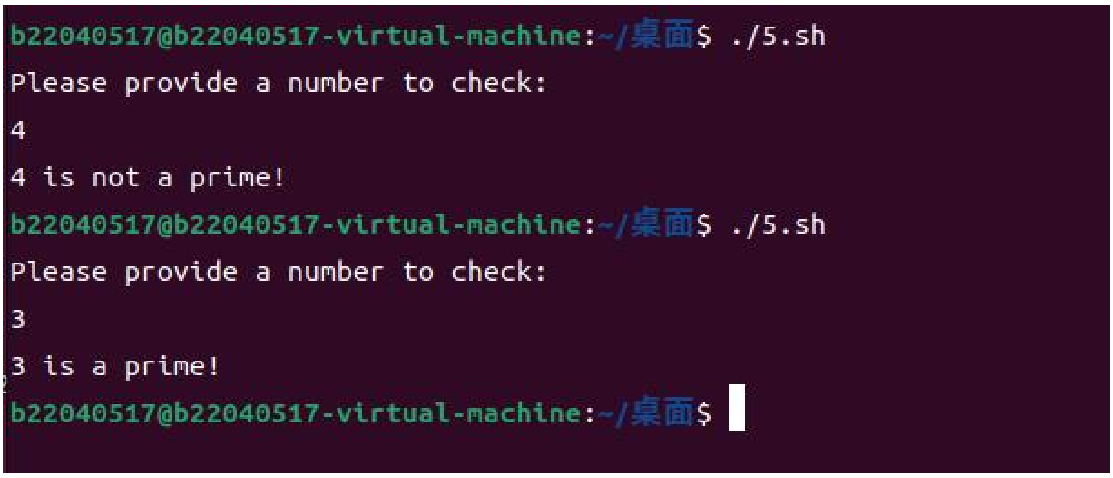


```bash
#!/bin/bash

prime() {
  local num=$1
  local flag=1
  local j=2
  while [ $j -le $((num / 2)) ]
  do
    if [ $((num % j)) -eq 0 ]
    then
      flag=0
      break
    fi
    j=$((j + 1))
  done
  echo $flag
}

if [ -z "$1" ]; then
  echo "Please provide a number to check:"
  read input
else
  input=$1
fi

result=$(prime $input)
if [ $result -eq 1 ]
then
  echo "$input is a prime!"
else
  echo "$input is not a prime!"
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$ 
```