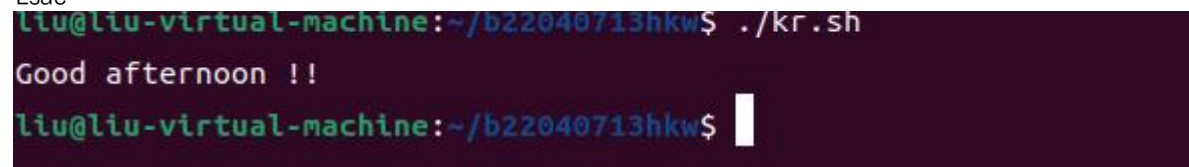


(1)

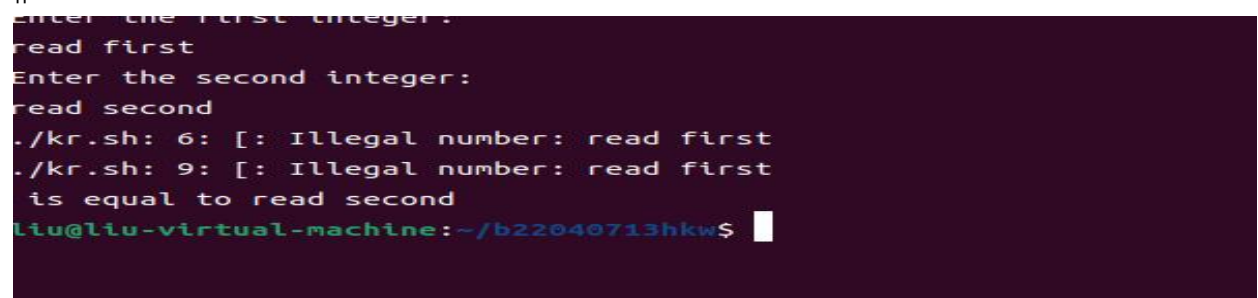
```
#!/bin/bash
hour = `date +%H`
case $hour in
  0[1-9] | 1[01] )
    echo "Good morining !!"
    ;;
  1[234567] )
    echo "Good afternoon !!"
    ;;
  * )
    echo "Good evening !! "
    ;;
Esac
```



A terminal window with a dark purple background. The prompt is `liu@liu-virtual-machine:~/b22040713hkw$`. The user has entered `./kr.sh`. The output is `Good afternoon !!`. The prompt is now `liu@liu-virtual-machine:~/b22040713hkw$` with a cursor.

(2)

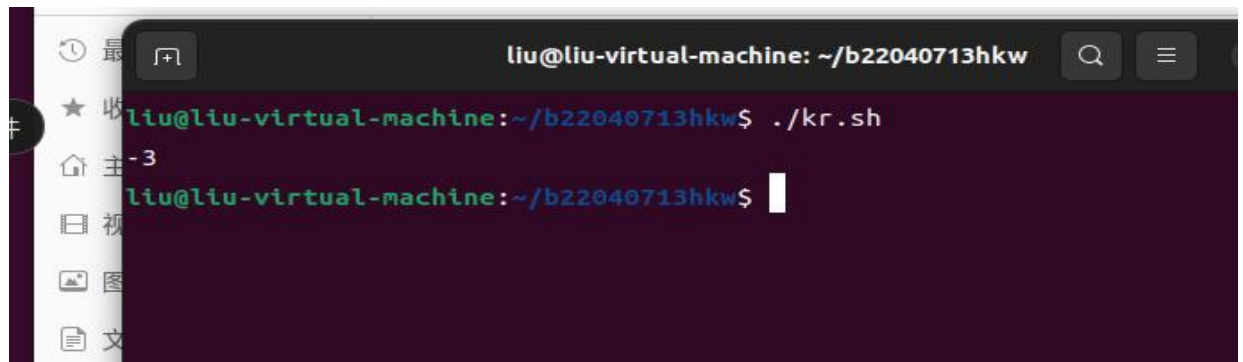
```
#!/bin/sh
echo "Enter the first integer:"
read first
echo "Enter the second integer:"
read second
if [ "$first" -gt "$second" ]
then
  echo "$first is greater than $second"
elif [ "$first" -lt "$second" ]
then
  echo "$FIRST is less than $second"
else
  echo "$FIRST is equal to $second"
fi
```



A terminal window with a dark purple background. The prompt is `liu@liu-virtual-machine:~/b22040713hkw$`. The user has entered `enter the first integer:`. The prompt is now `read first`. The user has entered `Enter the second integer:`. The prompt is now `read second`. The user has entered `6`. The prompt is now `./kr.sh: 6: [: Illegal number: read first`. The user has entered `9`. The prompt is now `./kr.sh: 9: [: Illegal number: read first`. The user has entered `is equal to read second`. The prompt is now `liu@liu-virtual-machine:~/b22040713hkw$` with a cursor.

(3)

```
#!/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 terminal window titled "liu@liu-virtual-machine: ~/b22040713hkw". The terminal shows a shell script being executed. The prompt is "liu@liu-virtual-machine:~/b22040713hkw\$". The first command is "./kr.sh", which outputs "-3". The second prompt is "liu@liu-virtual-machine:~/b22040713hkw\$". On the left side of the terminal, there is a sidebar with icons for "最近" (Recent), "收藏夹" (Favorites), "主目录" (Home), "视图" (View), "图片" (Images), and "文档" (Documents).

```
liu@liu-virtual-machine: ~/b22040713hkw
liu@liu-virtual-machine:~/b22040713hkw$ ./kr.sh
-3
liu@liu-virtual-machine:~/b22040713hkw$
```

(4)

```
#!/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
```

```
liu@liu-virtual-machine: ~/b22040713hkw
liu@liu-virtual-machine:~/b22040713hkw$ ./kr.sh
Total of 1 files executable
liu@liu-virtual-machine:~/b22040713hkw$
```

(5)

prime()

{

flag=1

j=2

while [\$j -le `expr \$1 / 2`]

do

if [`expr \$1 % \$j` -eq 0]

then

flag=0

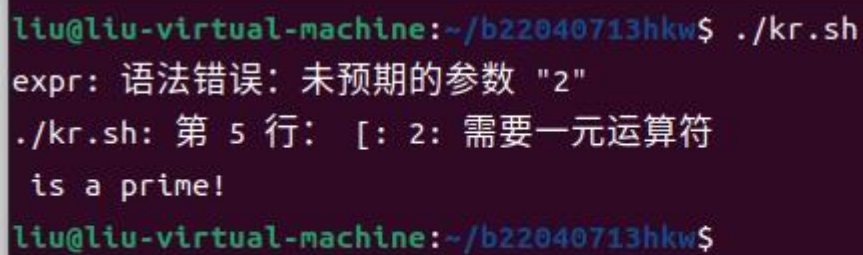
break

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!"
```

A terminal window with a dark purple background. The prompt is 'liu@liu-virtual-machine:~/b22040713hkw\$'. The user runs './kr.sh'. The output shows two error messages: 'expr: 语法错误: 未预期的参数 "2"' and './kr.sh: 第 5 行: [: 2: 需要一元运算符'. Then, the output 'is a prime!' is displayed. The prompt returns to 'liu@liu-virtual-machine:~/b22040713hkw\$'.

```
liu@liu-virtual-machine:~/b22040713hkw$ ./kr.sh
expr: 语法错误: 未预期的参数 "2"
./kr.sh: 第 5 行: [: 2: 需要一元运算符
is a prime!
liu@liu-virtual-machine:~/b22040713hkw$
```

f
i