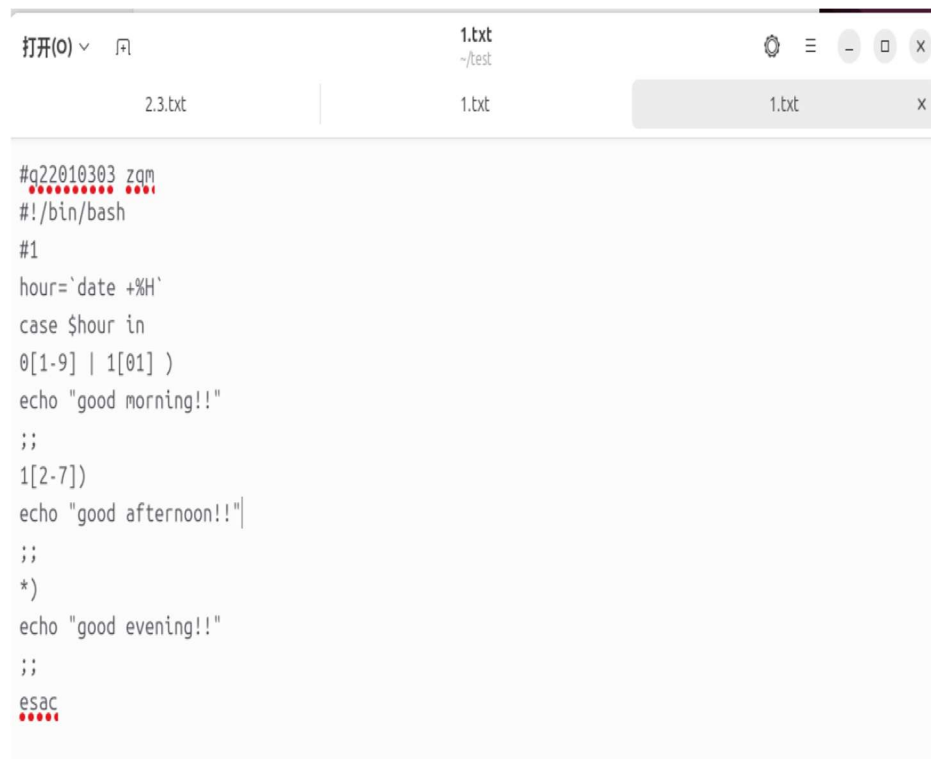
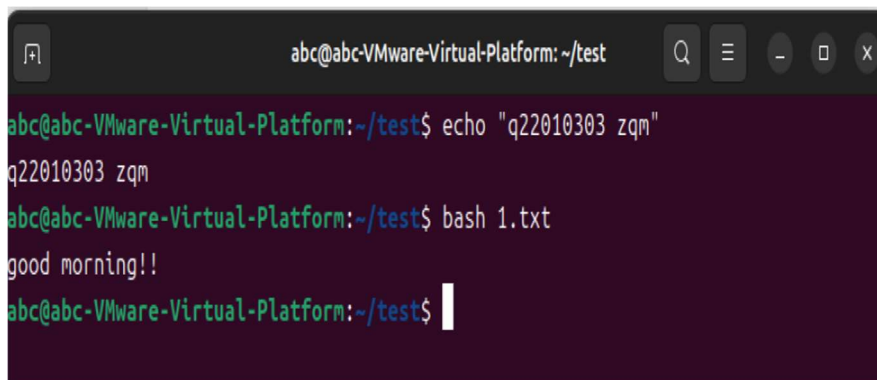


- 1 . Obtain the system time, and check whether it is in the morning, afternoon, or evening.



A screenshot of a file editor window. The title bar shows '打开(0)' and icons for search, save, and window management. The editor has two tabs: '2.3.txt' and '1.txt'. The '1.txt' tab is active, showing a shell script. The script starts with a comment '#q22010303 zqm', followed by '#!/bin/bash' and '#1'. It then uses a 'case' statement to check the hour of the day. If the hour is between 0 and 9, it prints 'good morning!!'. If the hour is between 12 and 7, it prints 'good afternoon!!'. Otherwise, it prints 'good evening!!'. The script ends with 'esac'.

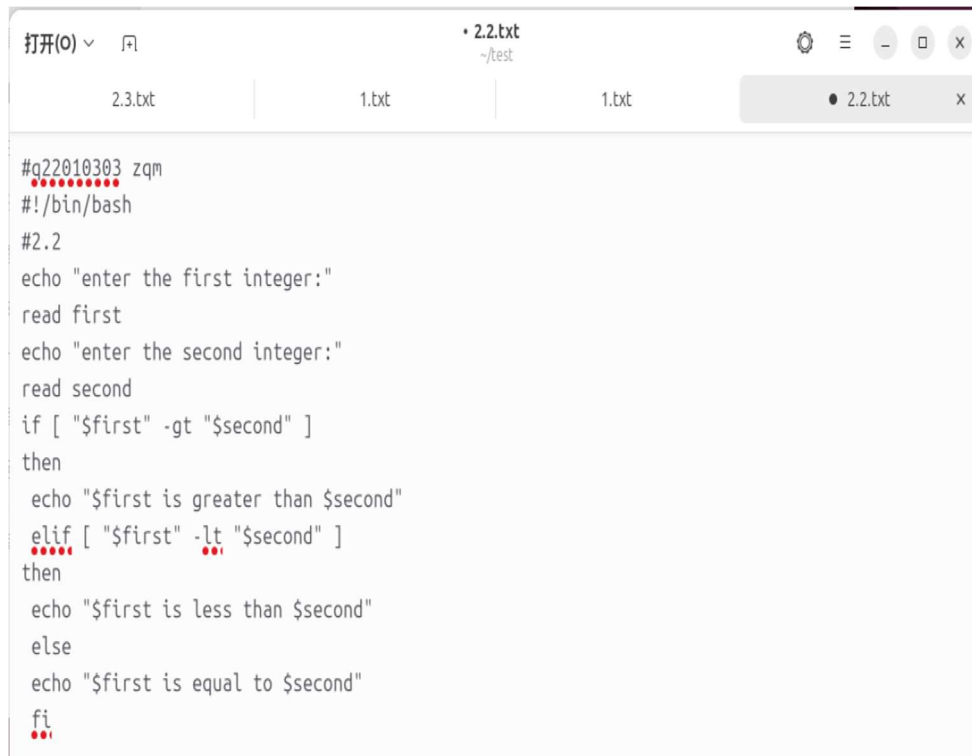
```
#q22010303 zqm
#!/bin/bash
#1
hour=`date +%H`
case $hour in
0[1-9] | 1[01] )
echo "good morning!!"
;;
1[2-7])
echo "good afternoon!!"
;;
*)
echo "good evening!!"
;;
esac
```



A screenshot of a terminal window. The title bar shows 'abc@abc-VMware-Virtual-Platform: ~/test' and icons for search, save, and window management. The terminal shows the following commands and output:

```
abc@abc-VMware-Virtual-Platform:~/test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test$ bash 1.txt
good morning!!
abc@abc-VMware-Virtual-Platform:~/test$
```

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



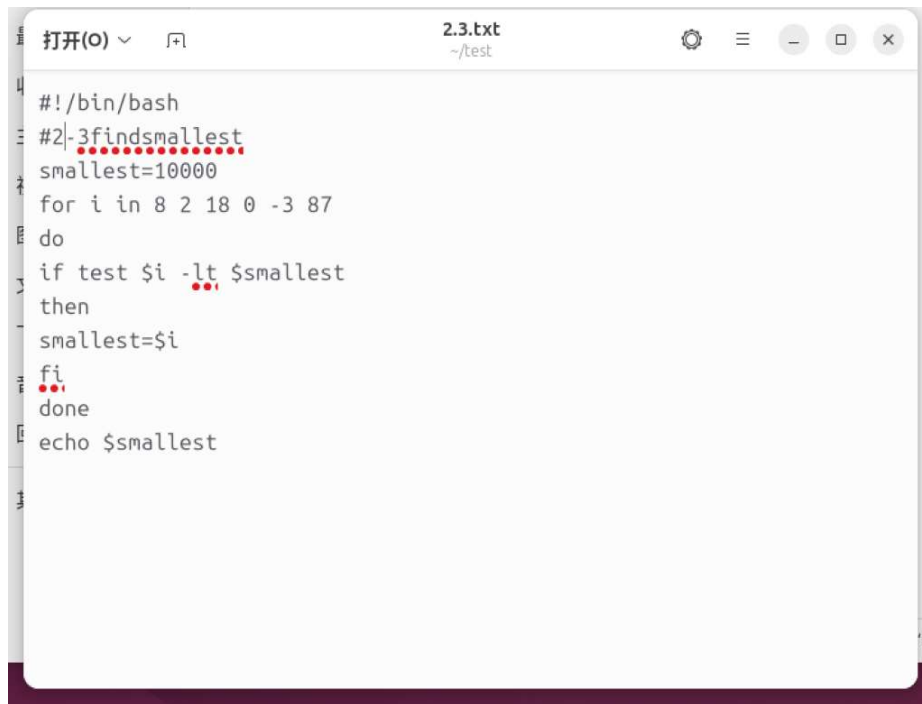
```
打开(O) 2.2.txt ~/test
2.3.txt 1.txt 1.txt 2.2.txt x

#q22010303 zqm
#!/bin/bash
#2.2
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
```



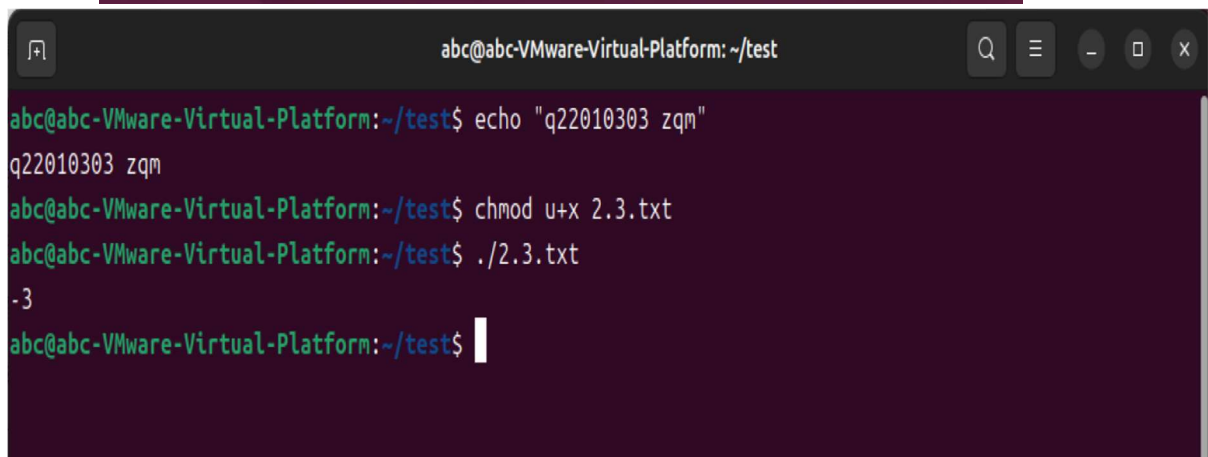
```
abc@abc-VMware-Virtual-Platform: ~/test
abc@abc-VMware-Virtual-Platform:~/test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test$ bash 2.2.txt
enter the first integer:
13
enter the second integer:
2
13 is greater than 2
abc@abc-VMware-Virtual-Platform:~/test$
```

3 . Find the minimal value in a given list.



A screenshot of a text editor window titled '2.3.txt' with the path '~/test'. The editor contains a shell script with the following content:

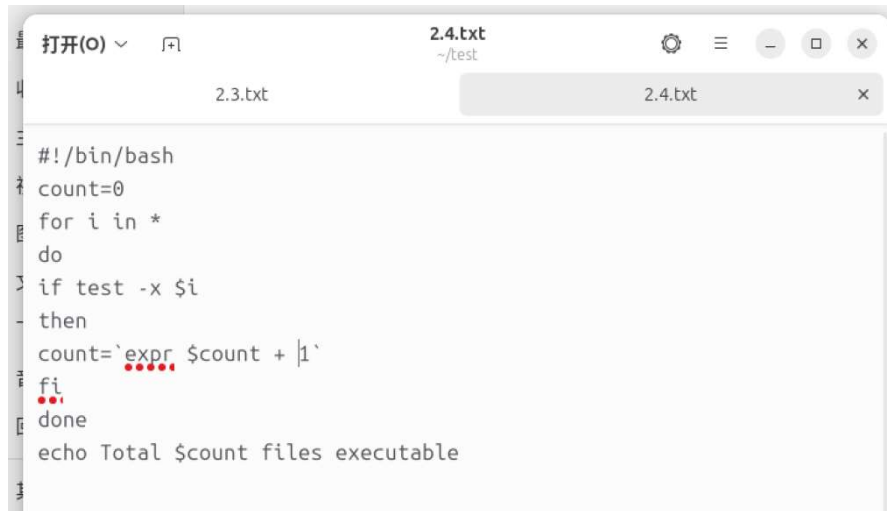
```
#!/bin/bash
#2-3findsmallest
smallest=10000
for i in 8 2 18 0 -3 87
do
if test $i -lt $smallest
then
smallest=$i
fi
done
echo $smallest
```



A screenshot of a terminal window with the title bar 'abc@abc-VMware-Virtual-Platform: ~/test'. The terminal shows the following commands and their outputs:

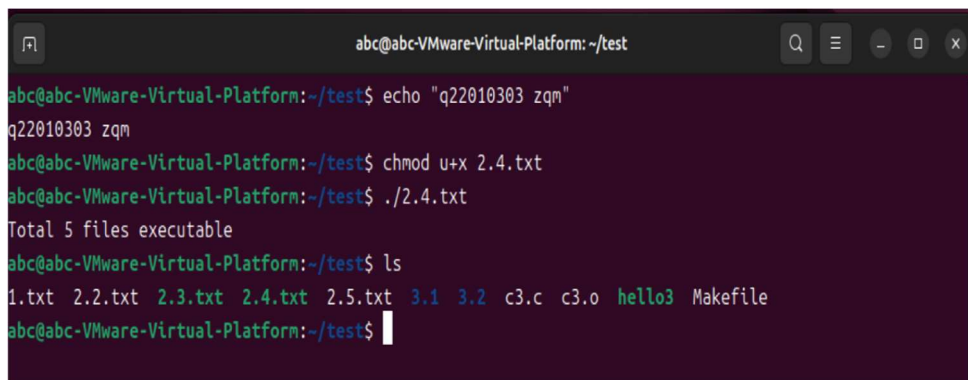
```
abc@abc-VMware-Virtual-Platform:~/test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test$ chmod u+x 2.3.txt
abc@abc-VMware-Virtual-Platform:~/test$ ./2.3.txt
-3
abc@abc-VMware-Virtual-Platform:~/test$
```

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



A screenshot of a text editor window titled '2.4.txt' with a subtitle '~/.test'. The editor contains a shell script designed to count the number of executable files in the current directory. The script uses a 'for' loop to iterate through all files, a 'test' command to check if each file is executable, and an 'expr' command to increment a counter. The final output is printed using 'echo'.

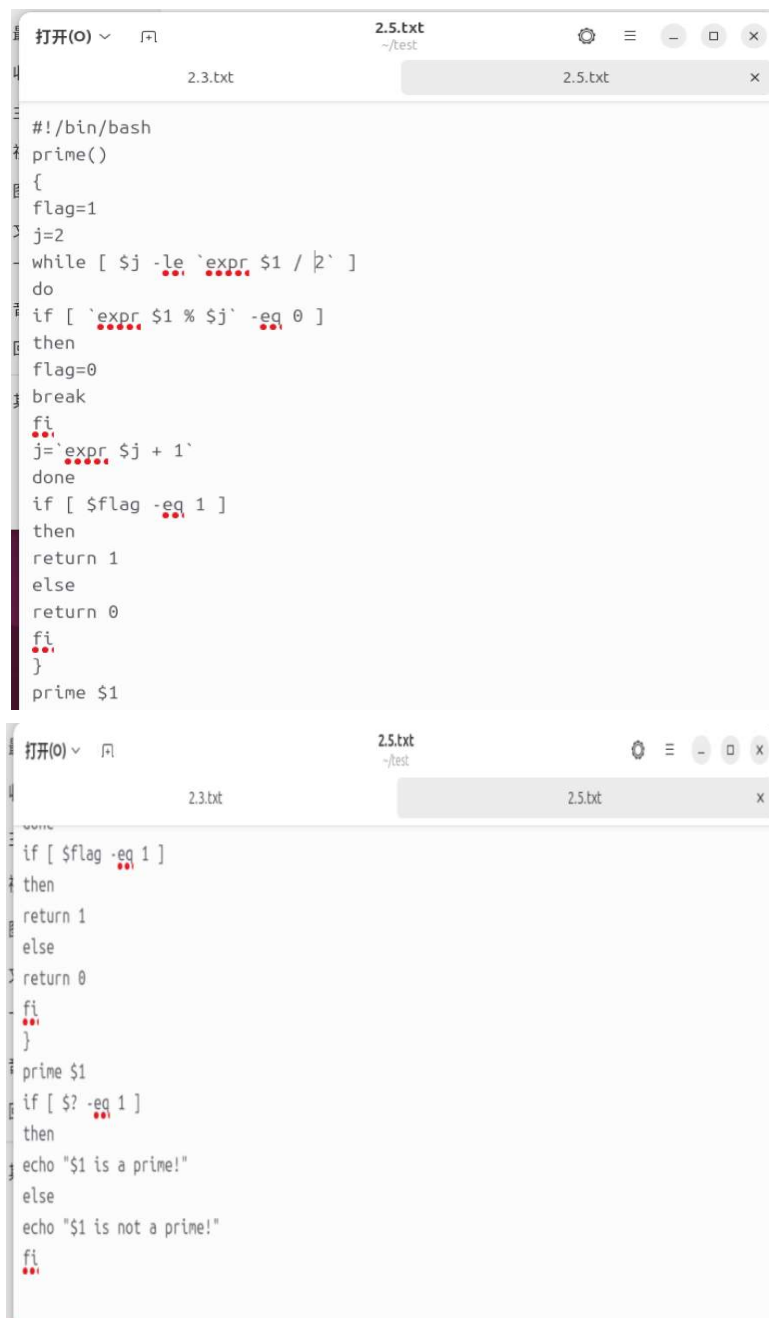
```
#!/bin/bash
count=0
for i in *
do
if test -x $i
then
count=`expr $count + 1`
fi
done
echo Total $count files executable
```



A screenshot of a terminal window with the title 'abc@abc-VMware-Virtual-Platform: ~/.test'. The terminal shows the execution of several commands: an echo command, a chmod command to set permissions on '2.4.txt', and the execution of the script './2.4.txt'. The script outputs 'Total 5 files executable'. Finally, the 'ls' command is run, displaying a list of files in the directory, including '1.txt', '2.2.txt', '2.3.txt', '2.4.txt', '2.5.txt', '3.1', '3.2', 'c3.c', 'c3.o', 'hello3', and 'Makefile'.

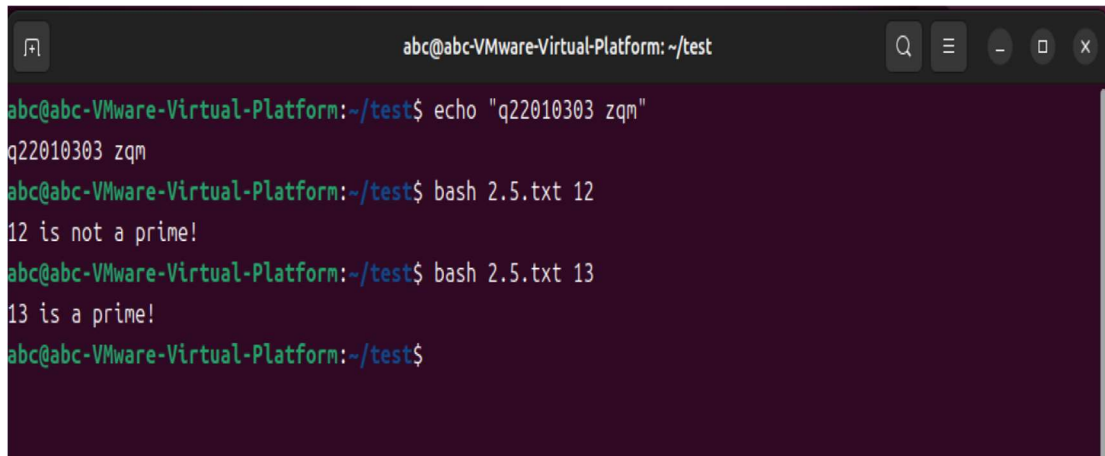
```
abc@abc-VMware-Virtual-Platform:~/.test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/.test$ chmod u+x 2.4.txt
abc@abc-VMware-Virtual-Platform:~/.test$ ./2.4.txt
Total 5 files executable
abc@abc-VMware-Virtual-Platform:~/.test$ ls
1.txt 2.2.txt 2.3.txt 2.4.txt 2.5.txt 3.1 3.2 c3.c c3.o hello3 Makefile
abc@abc-VMware-Virtual-Platform:~/.test$
```

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



```
#!/bin/bash
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 [ $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
```

A terminal window with a dark background. The title bar shows 'abc@abc-VMware-Virtual-Platform: ~/test'. The terminal contains the following text:

```
abc@abc-VMware-Virtual-Platform:~/test$ echo "q22010303 zqm"
q22010303 zqm
abc@abc-VMware-Virtual-Platform:~/test$ bash 2.5.txt 12
12 is not a prime!
abc@abc-VMware-Virtual-Platform:~/test$ bash 2.5.txt 13
13 is a prime!
abc@abc-VMware-Virtual-Platform:~/test$
```

Expr 表达式要用空格隔开每个项，一开始由于/和 2 之间没有加空格一直报错