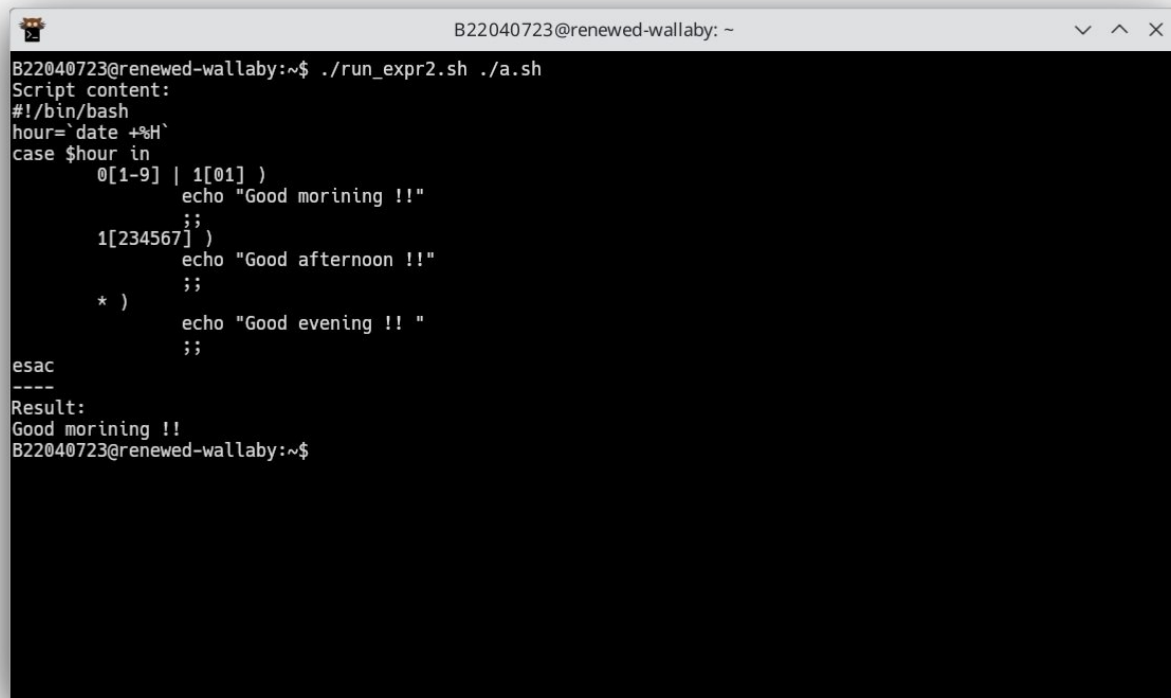


1. Obtain the system time, and check whether it is in the morning, afternoon, or evening.

A terminal window with a title bar showing the user 'B22040723@renewed-wallaby' and the current directory '~'. The terminal displays the execution of a script named 'a.sh' via 'run_expr2.sh'. The script's content is shown, including a shebang, a date assignment, and a case statement for time-based greetings. The output shows 'Good morining !!' (note the typo in the script).

```
B22040723@renewed-wallaby:~$ ./run_expr2.sh ./a.sh
Script content:
#!/bin/bash
hour=`date +%H`
case $hour in
    0[1-9] | 1[01] )
        echo "Good morining !!"
        ;;
    1[234567] )
        echo "Good afternoon !!"
        ;;
    * )
        echo "Good evening !! "
        ;;
esac
----
Result:
Good morining !!
B22040723@renewed-wallaby:~$
```

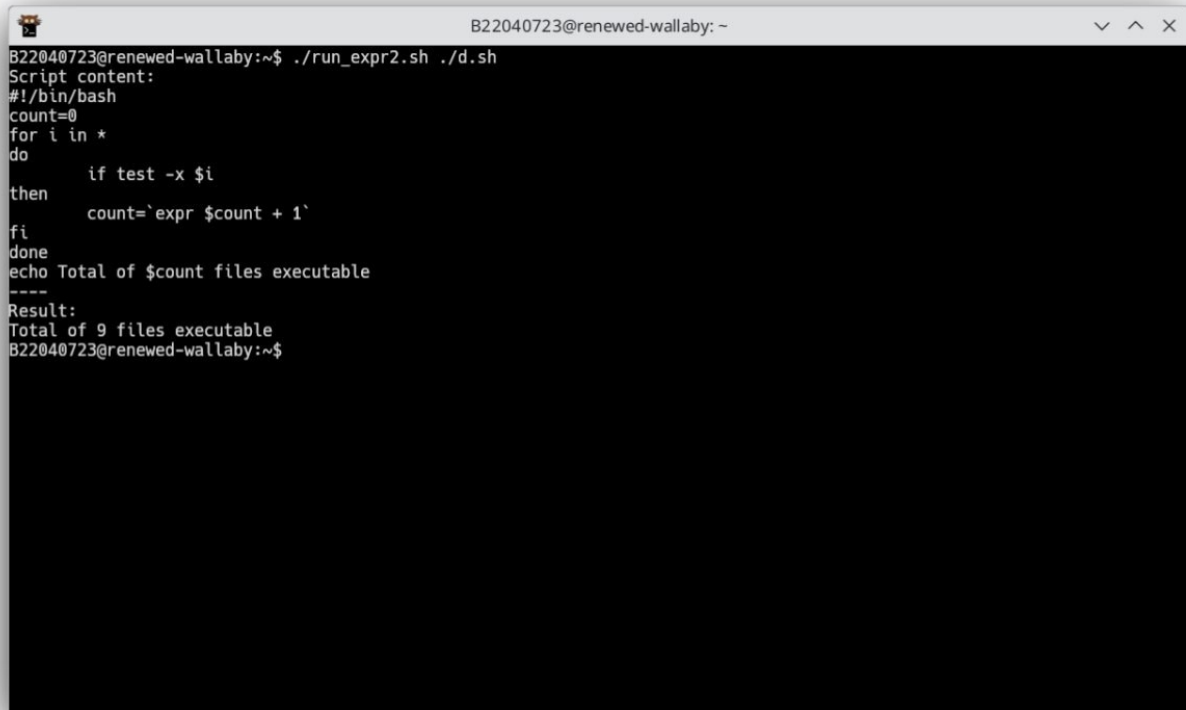
2. Input two number, check which one is greater, and output the result.

```
B22040723@renewed-wallaby: ~
B22040723@renewed-wallaby:~$ vim b.sh
B22040723@renewed-wallaby:~$ ./run_expr2.sh ./b.sh
Script content:
#!/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 $second"
fi
----
Result:
Enter the first integer:
10
Enter the second integer:
11
10 is less than 11
B22040723@renewed-wallaby:~$ ./b.sh
Enter the first integer:
22
Enter the second integer:
22
22 is equal to 22
B22040723@renewed-wallaby:~$ ./b.sh
Enter the first integer:
90
Enter the second integer:
1
90 is greater than 1
B22040723@renewed-wallaby:~$
```

3. Find the minimal value in a given list.

```
B22040723@renewed-wallaby: ~
B22040723@renewed-wallaby:~$ ./run_expr2.sh ./c.sh
Script content:
#!/bin/bash
smallest=10000
for i in 8 2 18 0 -3 87
do
    if [ $i -lt $smallest ]; then
        smallest=$i
    fi
done
echo $smallest
----
Result:
-3
B22040723@renewed-wallaby:~$
```

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

A terminal window with a title bar showing the username 'B22040723@renewed-wallaby' and a tilde '~'. The terminal displays the execution of a script named 'run_expr2.sh' with argument './d.sh'. The script's content is shown: it sets 'count=0', loops through all files in the current directory, checks if each file is executable using 'test -x \$i', and increments 'count' if so. The script then echoes the total count. The output shows 'Total of 9 files executable'.

```
B22040723@renewed-wallaby:~$ ./run_expr2.sh ./d.sh
Script content:
#!/bin/bash
count=0
for i in *
do
    if test -x $i
then
        count=`expr $count + 1`
    fi
done
echo Total of $count files executable
-----
Result:
Total of 9 files executable
B22040723@renewed-wallaby:~$
```

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

```
B22040723@renewed-wallaby: ~  
B22040723@renewed-wallaby:~$ ./run_expr2.sh ./e.sh 91  
Script content:  
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  
  
----  
Result:  
91 is not a prime!  
B22040723@renewed-wallaby:~$
```