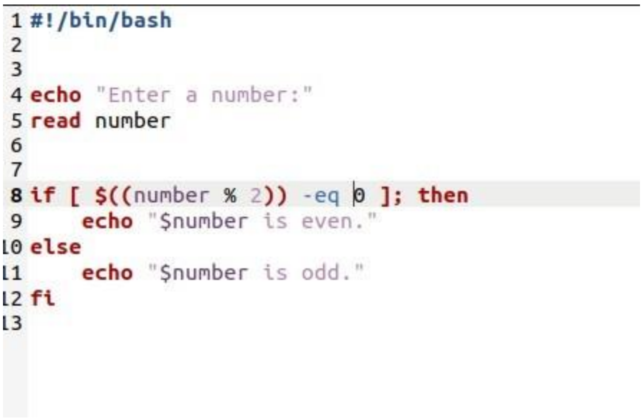
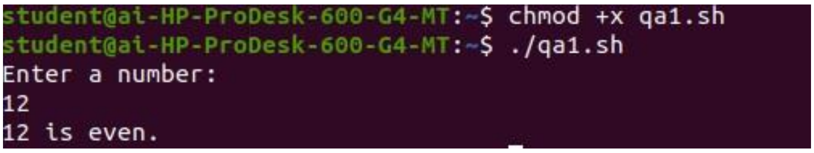
LAB TASK-4 423132 4THFEBRUARY

1. **Write a Shell program to check the given number is even or odd**

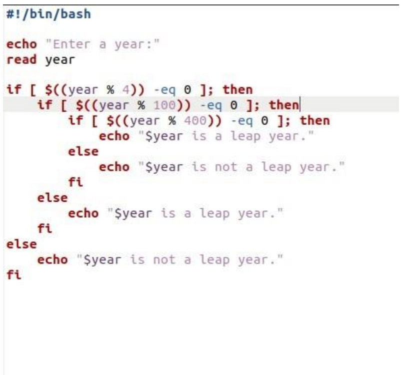


```
#!/bin/bash


echo "Enter a number:"
read number


if [ $((number % 2)) -eq 0 ]; then
    echo "$number is even."
else
    echo "$number is odd."
fi

```

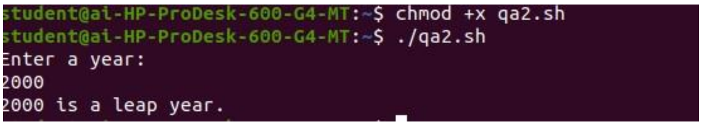


```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa1.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa1.sh
Enter a number:
12
12 is even.
```

2. **Write a Shell program to check the given year is leap year or not**

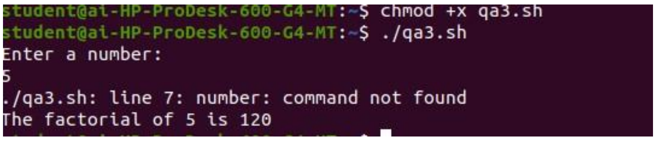


```
#!/bin/bash

echo "Enter a year:"
read year

if [ $((year % 4)) -eq 0 ]; then
    if [ $((year % 100)) -eq 0 ]; then
        if [ $((year % 400)) -eq 0 ]; then
            echo "$year is a leap year."
        else
            echo "$year is not a leap year."
        fi
    else
        echo "$year is a leap year."
    fi
else
    echo "$year is not a leap year."
fi
```



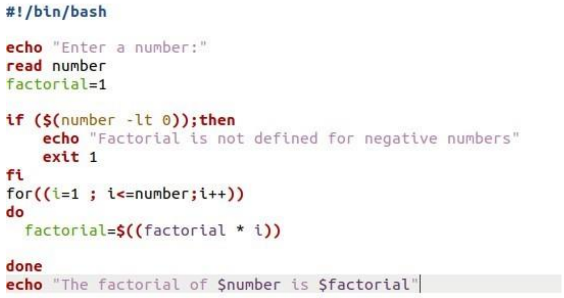
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa2.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa2.sh
Enter a year:
2000
2000 is a leap year.
```

3. **Write a Shell program to find the factorial of a number**



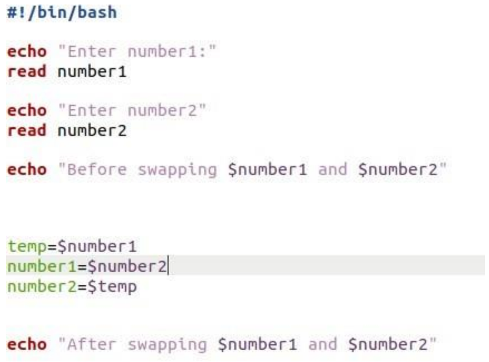
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa3.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa3.sh
Enter a number:
5
./qa3.sh: line 7: number: command not found
The factorial of 5 is 120
```

LAB TASK-4 423132   4THFEBRUARY

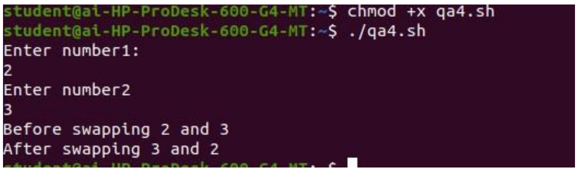


```
#!/bin/bash

echo "Enter a number:"
read number
factorial=1

if ($(number -lt 0));then
    echo "Factorial is not defined for negative numbers"
    exit 1
fi
for((i=1 ; i<=number;i++))
do
  factorial=$((factorial * i))

done
echo "The factorial of $number is $factorial"
```

4. **Write a Shell program to swap the two integers**

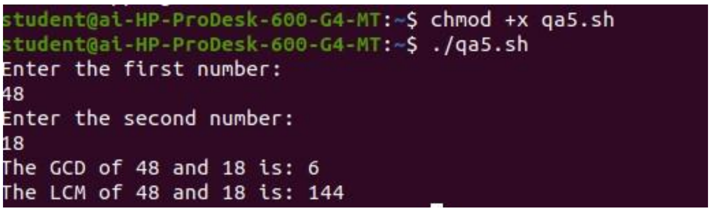


```
#!/bin/bash

echo "Enter number1:"
read number1

echo "Enter number2"
read number2

echo "Before swapping $number1 and $number2"


temp=$number1
number1=$number2
number2=$temp


echo "After swapping $number1 and $number2"
```



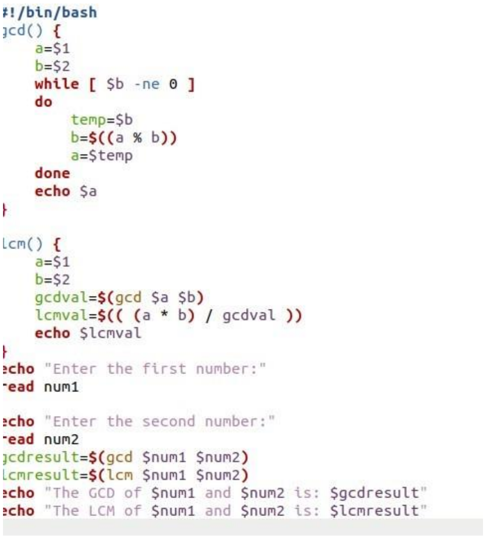
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa4.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa4.sh
Enter number1:
2
Enter number2
3
Before swapping 2 and 3
After swapping 3 and 2
```

5. Write a shell script to compute GCD & LCM of two numbers.



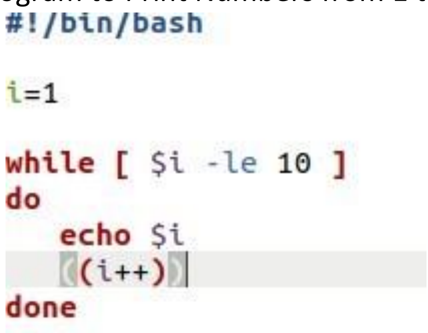
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa5.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa5.sh
Enter the first number:
48
Enter the second number:
18
The GCD of 48 and 18 is: 6
The LCM of 48 and 18 is: 144
```

LAB TASK-4 423132 4THFEBRUARY

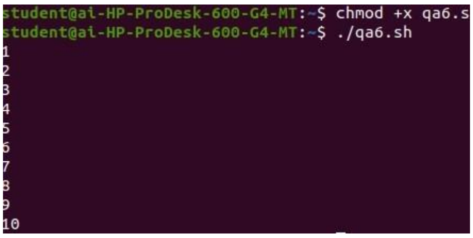


```
#!/bin/bash
gcd() {
    a=$1
    b=$2
    while [ $b -ne 0 ]
    do
        temp=$b
        b=$((a % b))
        a=$temp
    done
    echo $a
}

lcm() {
    a=$1
    b=$2
    gcdval=$(gcd $a $b)
    lcmval=$(( (a * b) / gcdval ))
    echo $lcmval
}
echo "Enter the first number:"
read num1

echo "Enter the second number:"
read num2
gcdresult=$(gcd $num1 $num2)
lcmresult=$(lcm $num1 $num2)
echo "The GCD of $num1 and $num2 is: $gcdresult"
echo "The LCM of $num1 and $num2 is: $lcmresult"
```

6. Shell Program to Print Numbers from 1 to 10 using While Loop

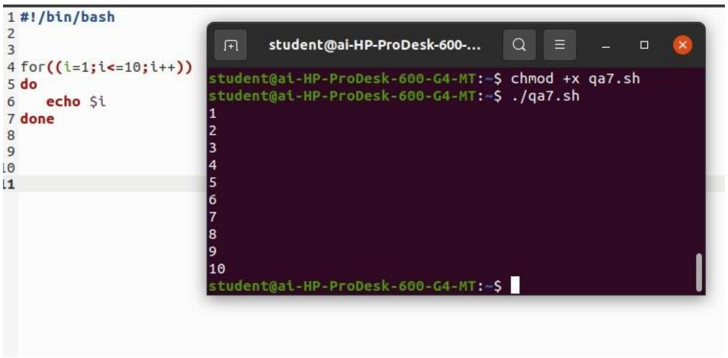


```
#!/bin/bash

i=1

while [ $i -le 10 ]
do
    echo $i
    ((i++))
done
```



```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa6.s
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa6.sh
1
2
3
4
5
6
7
8
9
10
```

7. Shell Program to Print Numbers from 1 to 10 using For Loop

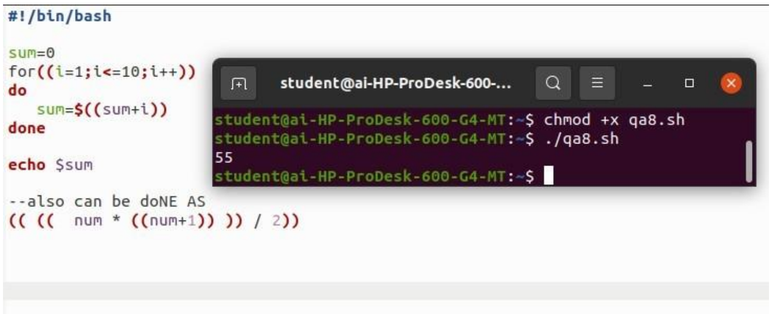


```
#!/bin/bash


for((i=1;i<=10;i++))
do
    echo $i
done
```

```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa7.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa7.sh
1
2
3
4
5
6
7
8
9
10
student@ai-HP-ProDesk-600-G4-MT:~$
```

8. write a shell script to find the sum of n numbers

LAB TASK-4 423132 4THFEBRUARY

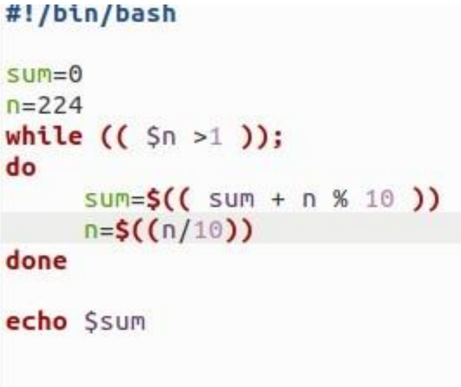


```
#!/bin/bash

sum=0
for((i=1;i<=10;i++))
do
   sum=$((sum+i))
done

echo $sum

--also can be doNE AS
(( ((  num * ((num+1)) )) / 2))
```

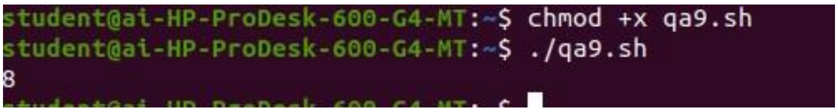
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa8.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa8.sh
55
student@ai-HP-ProDesk-600-G4-MT:~$ 
```

9. write a shell program for finding the sum of digits of a given number.

Sum = 1 + 2 + 3 + 4 +….+ N

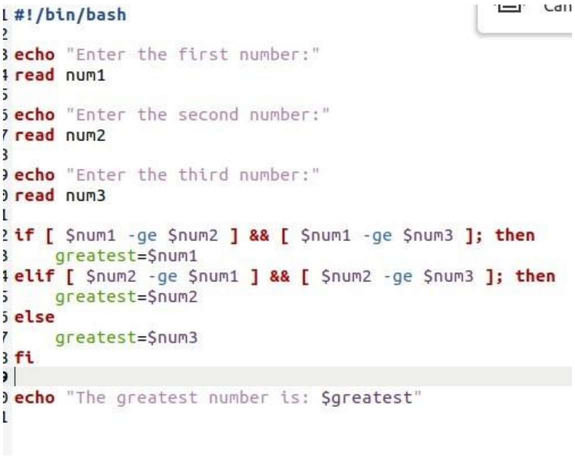


```
#!/bin/bash

sum=0
n=224
while (( $n >1 ));
do
      sum=$(( sum + n % 10 ))
      n=$((n/10))
done

echo $sum
```

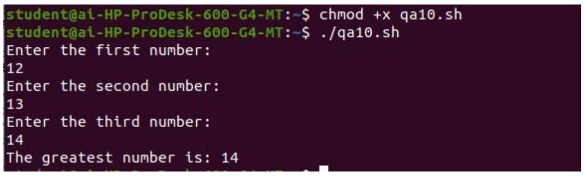


```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa9.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa9.sh
8
```

10. write a shell program for finding the greatest among three numbers.



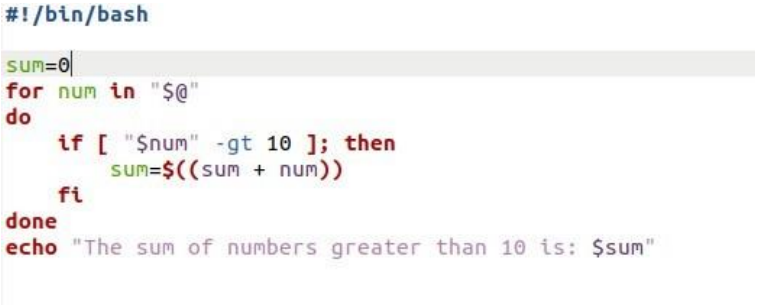
```
#!/bin/bash

echo "Enter the first number:"
read num1

echo "Enter the second number:"
read num2

echo "Enter the third number:"
read num3

if [ $num1 -ge $num2 ] && [ $num1 -ge $num3 ]; then
    greatest=$num1
elif [ $num2 -ge $num1 ] && [ $num2 -ge $num3 ]; then
    greatest=$num2
else
    greatest=$num3
fi

echo "The greatest number is: $greatest"
```



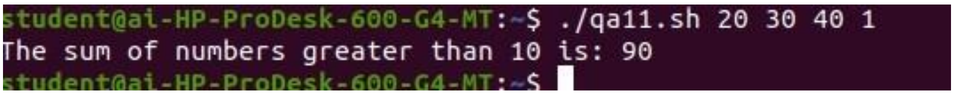
```
student@ai-HP-ProDesk-600-G4-MT:~$ chmod +x qa10.sh
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa10.sh
Enter the first number:
12
Enter the second number:
13
Enter the third number:
14
The greatest number is: 14
```

LAB TASK-4 423132 4THFEBRUARY

11. Write a shell script sum.sh that takes an unspecified number of command line arguments (up to 9) of ints and finds their sum. Modify the code to add a number to the sum only if the number is greater than 10



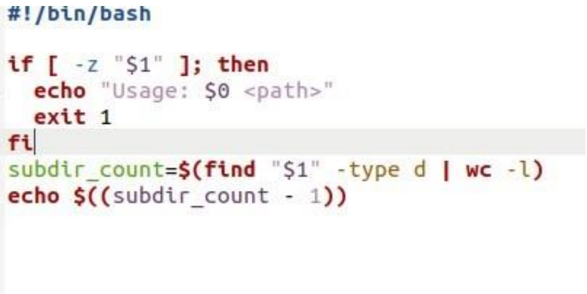
```
#!/bin/bash

sum=0
for num in "$@"
do
    if [ "$num" -gt 10 ]; then
        sum=$((sum + num))
    fi
done
echo "The sum of numbers greater than 10 is: $sum"
```



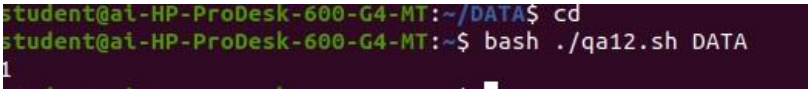
```
student@ai-HP-ProDesk-600-G4-MT:~$ ./qa11.sh 20 30 40 1
The sum of numbers greater than 10 is: 90
student@ai-HP-ProDesk-600-G4-MT:~$
```

12. Write a shell script takes the name a path (eg: /afs/andrew/course/15/123/handin), and counts all the sub directories (recursively).

if the path is provided as an argument

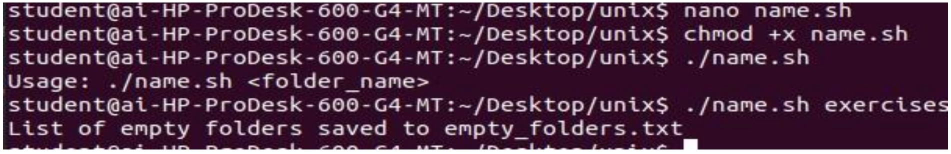


```
#!/bin/bash

if [ -z "$1" ]; then
  echo "Usage: $0 <path>"
  exit 1
fi
subdir_count=$(find "$1" -type d | wc -l)
echo $((subdir_count - 1))
```

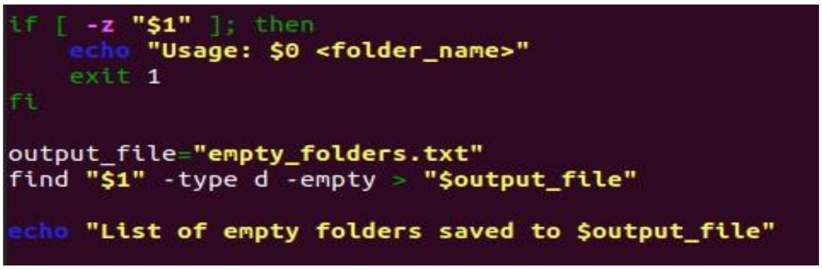


```
student@ai-HP-ProDesk-600-G4-MT:~/DATA$ cd
student@ai-HP-ProDesk-600-G4-MT:~$ bash ./qa12.sh DATA
1
```

13. Write a shell script that takes a name of a folder as a command line argument, and produce a file that contains the names of all sub folders with size 0 (that is empty sub folders)



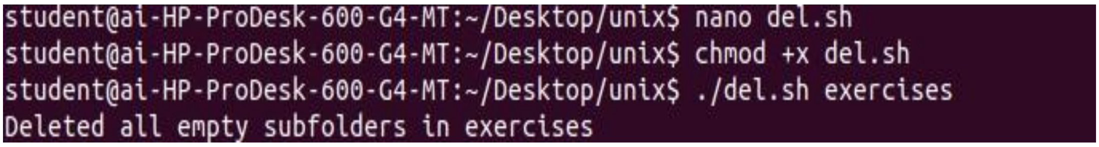
```
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ nano name.sh
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ chmod +x name.sh
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ ./name.sh
Usage: ./name.sh <folder_name>
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ ./name.sh exercises
List of empty folders saved to empty_folders.txt
```



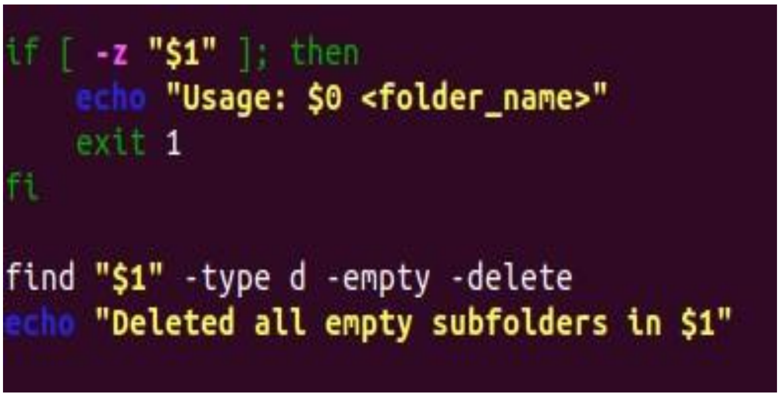
```
if [ -z "$1" ]; then
    echo "Usage: $0 <folder_name>"
    exit 1
fi

output_file="empty_folders.txt"
find "$1" -type d -empty > "$output_file"

echo "List of empty folders saved to $output_file"
```

LAB TASK-4 423132 4THFEBRUARY

14. Write a shell script that takes a name of a folder, and delete all sub folders of size 0

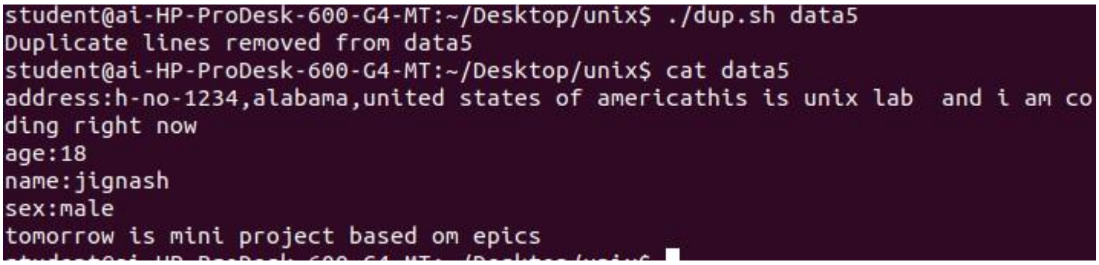


```
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ nano del.sh
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ chmod +x del.sh
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ ./del.sh exercises
Deleted all empty subfolders in exercises
```

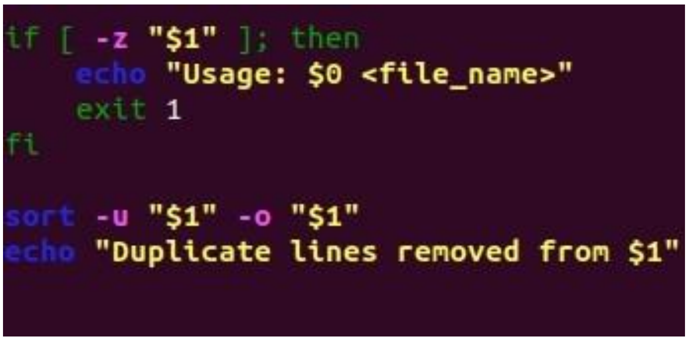


```
if [ -z "$1" ]; then
    echo "Usage: $0 <folder_name>"
    exit 1
fi

find "$1" -type d -empty -delete
echo "Deleted all empty subfolders in $1"
```

15.write a shell script that will take an input file and remove identical lines (or duplicate lines from the file



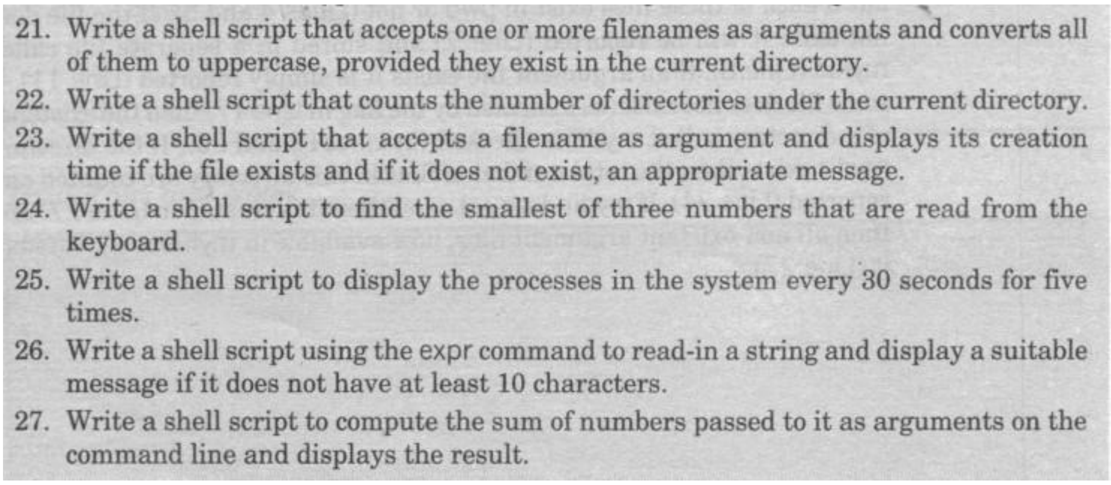
```
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ ./dup.sh data5
Duplicate lines removed from data5
student@ai-HP-ProDesk-600-G4-MT:~/Desktop/unix$ cat data5
address:h-no-1234,alabama,united states of americathis is unix lab  and i am co
ding right now
age:18
name:jignash
sex:male
tomorrow is mini project based om epics
```



```
if [ -z "$1" ]; then
    echo "Usage: $0 <file_name>"
    exit 1
fi

sort -u "$1" -o "$1"
echo "Duplicate lines removed from $1"
```

LAB TASK-4 423132 4THFEBRUARY

16.Complete the following exercises:



21. Write a shell script that accepts one or more filenames as arguments and converts all of them to uppercase, provided they exist in the current directory.
22. Write a shell script that counts the number of directories under the current directory.
23. Write a shell script that accepts a filename as argument and displays its creation time if the file exists and if it does not exist, an appropriate message.
24. Write a shell script to find the smallest of three numbers that are read from the keyboard.
25. Write a shell script to display the processes in the system every 30 seconds for five times.
26. Write a shell script using the expr command to read-in a string and display a suitable message if it does not have at least 10 characters.
27. Write a shell script to compute the sum of numbers passed to it as arguments on the command line and displays the result.