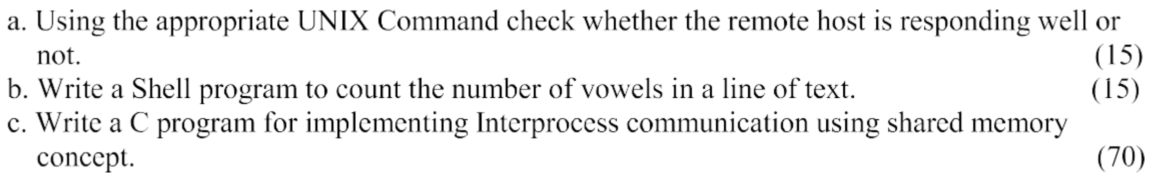
Important qp for os lab

4. 

a. Using the appropriate UNIX Command check whether the remote host is responding well or not. (15)
b. Write a Shell program to count the number of vowels in a line of text. (15)
c. Write a C program for implementing Interprocess communication using shared memory concept. (70)

a)

Using the appropriate Unix command, check whether the remote host is responding well or not.

Command Used:

```
bash
ping <remote_host>
```

Example:

```
bash
ping google.com
```

Explanation:

- The `ping` command sends Internet Control Message Protocol (ICMP) echo request packets to the specified host.
- If the host is responding well, you will receive ICMP echo replies along with time statistics (in milliseconds).
- If the host is **not** responding, you may see messages like:
  - `Request timed out`
  - `Destination Host Unreachable`
  - No reply at all.

Sample Output (if responding):

```
PING google.com (142.250.64.142): 56 data bytes
64 bytes from 142.250.64.142: icmp_seq=0 ttl=117 time=14.3 ms
64 bytes from 142.250.64.142: icmp_seq=1 ttl=117 time=13.7 ms
...
```

b)

```
#!/bin/bash

echo "Enter a line of text:"

read text
```

```
text=$(echo "$text" | tr 'A-Z' 'a-z')

count=0


for (( i=0; i<${#text}; i++ )); do

   char=${text:$i:1}

   case $char in

      a|e|i|o|u)

         count=$((count + 1))

         ;;

   esac

done

echo "Number of vowels: $count"
```

✅ **Sample Output**

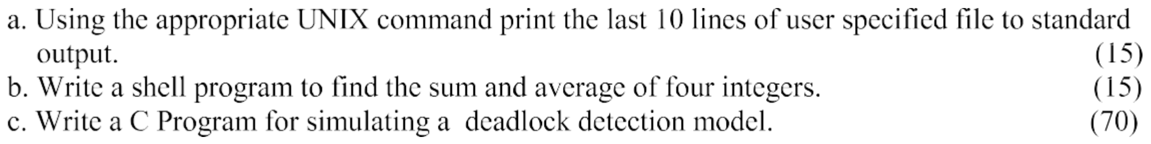
```
bash

Enter a line of text:

Operating Systems Lab

Number of vowels: 7
```

5.



a. Using the appropriate UNIX command print the last 10 lines of user specified file to standard output. (15)
b. Write a shell program to find the sum and average of four integers. (15)
c. Write a C Program for simulating a deadlock detection model. (70)

```
a)echo "Enter the filename:"

read filename

tail -n 10 "$filename"
```

□ Sample Output:

```
Enter the filename:

sample.txt

(line 91)

(line 92)
```

```
...
(line 100) echo "Enter the filename:"
```

b)

```
echo "Enter first number:"
read a
echo "Enter second number:"
read b
echo "Enter third number:"
read c
echo "Enter fourth number:"
read d
sum=$((a + b + c + d))
average=$(echo "scale=2; $sum / 4" | bc)
echo "Sum = $sum"
echo "Average = $average"
```

▯ **Sample Output:**

bash

```
Enter first number:
10
Enter second number:
20
Enter third number:
30
Enter fourth number:
40
Sum = 100
Average = 25.00
```

7.

a. Using the desired UNIX command rename a file of the user to a new name.
b. Write a Shell program to find the area and circumference of a circle.
c. Write a C program for implementing Priority Scheduling algorithm.

a)
```
echo "Enter the current filename:"

read old_name

echo "Enter the new filename:"

read new_name

mv "$old_name" "$new_name"

echo "File renamed from $old_name to $new_name"
```

▯ **Sample Output:**

```
Enter the current filename:

report.txt

Enter the new filename:

final_report.txt

File renamed from report.txt to final_report.txt
```

B)

```
echo "Enter the radius of the circle:"

read radius

pi=3.1416

area=$(echo "scale=2; $pi * $radius * $radius" | bc)

circumference=$(echo "scale=2; 2 * $pi * $radius" | bc)

echo "Area of the circle = $area"

echo "Circumference of the circle = $circumference"
```

▯ **Sample Output:**

```
Enter the radius of the circle:

5

Area of the circle = 78.54

Circumference of the circle = 31.42
```

9.

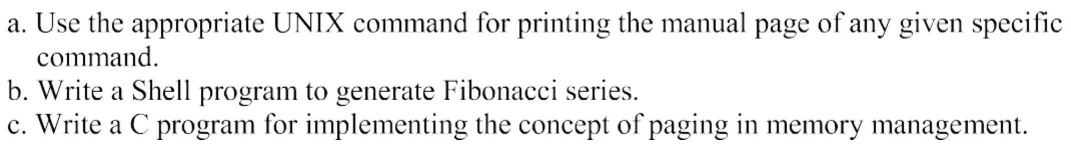
a. Write a Shell program to display student grades.
b. Write a C program for implementing the conept of synchronization in threads.

```
a) echo "Enter student's marks (0–100):"
read marks
if [ "$marks" -ge 0 ] && [ "$marks" -le 100 ]; then
   if [ "$marks" -ge 90 ]; then
      grade="A"
   elif [ "$marks" -ge 80 ]; then
      grade="B"
   elif [ "$marks" -ge 70 ]; then
      grade="C"
   elif [ "$marks" -ge 60 ]; then
      grade="D"
   elif [ "$marks" -ge 50 ]; then
      grade="E"
   else
      grade="F (Fail)"
   fi
 echo "Student Grade: $grade"
else
   echo "Invalid marks. Please enter a value between 0 and 100."
fi
```

▯ Sample Output:

```
Enter student's marks (0–100):
78
Student Grade: C
```

10.


a. Use the appropriate UNIX command for printing the manual page of any given specific command.
b. Write a Shell program to generate Fibonacci series.
c. Write a C program for implementing the concept of paging in memory management.

a) To view the manual page for the `ls` command:

```
bash

man ls
```

🖥️ How It Works:

- The `man` command displays the **manual (help) page** for the specified command.
- This includes **usage**, **options**, **description**, and **examples**.
- Press `q` to quit the manual viewer.

Sample Output Snippet:

```
LS(1)                      User Commands                     LS(1)

NAME
       ls - list directory contents

SYNOPSIS
       ls [OPTION]... [FILE]...

DESCRIPTION
       List  information  about the FILEs (the current directory by default).
```

b) echo "Enter the number of terms in the Fibonacci series:"

```
read n

if [ "$n" -le 0 ]; then

  echo "Please enter a positive number."

  exit 1

fi
```

```
a=0

b=1

echo "Fibonacci Series up to $n terms:"

for (( i=0; i<n; i++ ))

do

   echo -n "$a "

   fn=$((a + b))

   a=$b

   b=$fn

done

echo
```

▯ **Sample Output:**

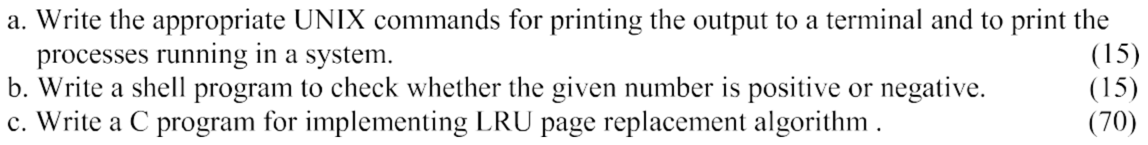
```
Enter the number of terms in the Fibonacci series:

7

Fibonacci Series up to 7 terms:

0 1 1 2 3 5 8
```

12.



a. Write the appropriate UNIX commands for printing the output to a terminal and to print the processes running in a system. (15)
b. Write a shell program to check whether the given number is positive or negative. (15)
c. Write a C program for implementing LRU page replacement algorithm . (70)

Print processes running on the system

Use the `ps` command to list currently running processes.

```
bash
```

By default, `ps` shows processes running in the current terminal session.

To see **all processes** running on the system, use:

```
bash
ps -ef
```

or

```
bash
ps aux
```

Summary:

Purpose	Command
Print output to terminal	`echo "your text"`
Show running processes	`ps`
Show all system processes	`ps -ef` or `ps aux`

b) Shell Program: Check Positive or Negative Number

```
bash
echo "Enter a number:"
read num

if ! [[ "$num" =~ ^-?[0-9]+$ ]]; then
    echo "Error: Please enter a valid integer."
    exit 1
fi
echo "The number $num is positive."
elif [ "$num" -lt 0 ]; then
    echo "The number $num is negative."
else
    echo "The number is zero."
fi
```

□ Sample Output 1 (Positive number):

```
Enter a number:
25
The number 25 is positive.
```

Sample Output 2 (Negative number):

```
csharp
CopyEdit
Enter a number:
-8
The number -8 is negative.
```

13.

a. Write a shell script to display the digits which are in odd position in a given number.
b. Write a C program for implementing Optimal page replacement algorithm.
a)

Shell Script: Display Digits in Odd Positions

```
bash
CopyEdit

echo "Enter a number:"
read number

length=${#number}
echo -n "Digits in odd positions: "

for (( i=0; i<length; i++ ))
do
    )
    if (( i % 2 == 0 )); then
        echo -n "${number:$i:1} "
    fi
done

echo
```

□ Sample Output:

```
Enter a number:
123456789
Digits in odd positions: 1 3 5 7 9
```

14.

a. Write a shell program to find the sum of two numbers using function programming
b. Write a shell program for implementing sequential file allocation strategy.

a) Sum of Two Numbers Using Function

```
bash
sum() {
    local a=$1
    local b=$2
    echo $((a + b))
}

echo "Enter first number:"
read num1
echo "Enter second number:"
read num2
```

```
result=$(sum $num1 $num2)

echo "Sum of $num1 and $num2 is: $result"
```

□ Sample Output:

```
Enter first number:
12
Enter second number:
30
Sum of 12 and 30 is: 42
```