

Department of Computer Science

Gujarat University



Certificate

Roll No: 26

Seat No: _____

This is to certify that Mr./Ms. Rutwik A. Patel
student of MCA Semester – III has duly completed his/her term work for
the semester ending in December 2020, in the subject of
OPERATING SYSTEMS towards partial fulfillment
of his/her Degree of Masters in Computer Applications.

10th-Dec-2020

Date of Submission

Internal Faculty

Head of Department

Department Of Computer Science
Rollwala Computer Centre
Gujarat University

MCA – III

Subject: - Operating System

Name: - Rutwik A. Patel

Roll No.: - 26

Exam Seat No.: -

OS Assignment

* Base address

An address that is used as the origin in the calculation of address in the execution of a program

* Batch Processing

Pertaining to the technique of executing a set of computer programs such that each is completed before the next program of the set is started.

* Binary Semaphore

- A semaphore that only takes on only values 0 and 1.
- Binary Semaphore always only one process or thread to have access to a shared critical resource at a time.

* Block

A collection of contiguous records.



that are recorded as a unit
the units are separated
by the interblock gaps.

2) A group of lists that are transmitted as a unit.

* B-tree

A technique for organizing indices in order to keep access time to a minimum it stores the data keys in a balanced hierarchy that continually and realigns itself as items are inserted and deleted thus all nodes always have a similar number of keys.

* Busy-Waiting

The repeated execution of loop of code while waiting for an event to occur.

* Cache Memory.

→ A memory that is smaller and faster than main memory and that is located between the Processor and main memory. The Cache acts as buffer for recently used memory locations.

* CPU

→ The part of a Computer that fetches and executes instructions. It consists of an arithmetic and logic unit, a control unit and registers often simply referred to as Processor.

* Cluster.

→ A group of interconnected whole computers working together as a unified computing resource that can create the illusion of being one machine.



the term whole Computer means a system that can run on its own apart from the cluster.

* Concurrent

→ Pertaining the Process or threads that take place within a common interval of time during which they may have to alternately share the resources.

* Consumable Resources

→ A resource that can be created or destroyed when a resource is acquired by a process. The resource ceases to exist. Examples are interrupts, signals, messages and information.

* Database

→ A collection of interrelated data often with controlled redundancy.

organized according to schema to serve one or more applications the data are stored so that they can be used by different programs without any structure or organization.

A common approach is used to add new data and to modify and retrieve existing data.

* Deadlock

→ An impasse that occurs when multiple processes are waiting for some other resource that will not become available because it is being held by another process that is in similar state.

→ This state is also known as circular wait since all process are waiting for another.

* Demand Paging

→ the transfer of Page from main memory to Secondary memory at the moment of need

* Device driver.

→ A module that deals directly with the I/O module.

* Direct access.

→ The capability to obtain data from a storage device or to enter data into a storage device in a sequence independent of their position by means of address that indicate physical location of data.

* Direct memory access.

→ A form of I/O which is a special module called a data module that controls

Exchange of the data between the main memory and I/O device the Processor sends request for transfer of a block of data. The DMA module is interrupted.

* Disable Interrupt

→ A condition usually created by the operating system during which the Processor will ignore the interrupt request signal of the specified class.

* Disk allocation table

→ A table that indicates which block of Secondary Storage are free and available for allocation of files.

* Distributed os.

→ A common operating system that is shared by a network of computers. The operating system provides support for interprocess communication and the prevention of deadlock.

* Dynamic allocation.

→ Process that assigns here is absolute address to a computer program during execution so that the program may be executed from different area of main storage.

* Enable interrupt.

→ Condition usually created by OS during which the processor will respond to.

interruption Signals of a Application Class.

* External fragmentation

→ Occurs when memory is divided into variable size of Partition Corresponding to the blocks of data assigned to the memory As Segments are moved into and out of the memory gaps will occur between occupied Partitions of the memory.

* file

→ A Set of Related Records Created as Units

* field

→ defined logical data that are Part of Record.

→ An Elementary Record may Contain a date item or



a pointer or a link.

* file allocation table

→ A table that indicates the physical location on secondary storage of the space allocated to a file.

* file Management System

→ A set of system software that provides services to users and applications in the use of files including file access directory maintenance.

* file organisation

→ The physical records of in a file as determined by access method used to store and retrieve them.

- * first in first out.
 - A queuing technique in which the next item to be retrieved is the item which was first inserted and so on.

- * first come first serve
 - Same as fifo.

- * hash file
 - A file in which records are accessed according to the values of key field.
 - hashing is used to locate a record on the basis of its key value.
 - hashing is used to locate a record on the basis of the its key value.

* Hashing

- The Selection of a storage location for item of data by calculating the address as a function of contents to the data.
- This technique complicates the storage allocation and results in a rapid random interval.

* Hit Ratio

- In a two level memory the fraction of memory access that are found in the faster memory.

* Indexed Access.

- Partitioning to the organization and access of the records of storage structure through a separate index to the location of stored records.



* Indexed file

- > A file in which records are accessed depending to the value of key fields.
- > An index is required that indicates the location of each row on the basis of each key value.

* Indexed Sequential access

- > Pertaining to the organization and accessing of the records of a storage of the record structure through the keys that are stored in the arbitrary Partitioned Sequential file.



- * Induced Sequential files.
 - A file that can in which records are ordered according to the values of key field.
 - The main file is supplemented with an index file that contains a partial list of key values. This index provides lookup facility to quickly reach the desired record.
- * Instruction Cycle.
 - The time Period during which an instruction is fetched and Executed in memory when a Computer is given instructions in the machine language.



* Internal fragmentation.

- Occurs When memory is divided into fixed number of Partitions.
- if a block of data is assigned to one or more Partitions then there may be wasted space in the last Partition.
- This will occur when the last Partition of data is smaller than the last Partition.

* Interrupt.

- & Suspension of a Process such as the Execution of Computer Programs caused by an Event to the Process and performed in a Such a Way the Process can be Resumed.

* Interrupt handler

→ It is generally part of the operating system when an interrupt occurs the control is transferred into corresponding handler which takes some action.

* Job

→ A set of steps packed into form as a same unit.

* Kernel

→ Portion of operating system that includes most heavily used portions of the software. Generally it is maintained permanently into the device's main memory.

→ The kernel runs in the privileged mode and responds to the most recent calls from device.

* Kernel Mode

→ A Privileged mode of Execution reserved for the kernel of operating System typically kernel mode allows access to regions of main memory that are not available to process executing in less Privileged mode.

→ it also enables execution of certain machine instructions that are restricted to the kernel mode also referred as system mode or kernel mode.

* LIFO

- A queuing technique in which the next item to be retrieved is the item most recently placed in the queue.

* LiveLock

- A condition in which two or more processes continuously change their state in response to other process without doing any useful work.
- This situation is similar to the deadlock of different process where in that also no progress is made, it differs in that neither process is blocked or waiting.

* Logical Address

- A reference to a memory location independent of the current assignment of data to the memory.
- A translation must be made to a Physical address before the memory access can be achieved.

* Logical Record

- A record independent of its Physical Environment of Partitions of logical Record may be located in the different Physical record or several record or Parts of logical Records stored as a part of many be located into the one Physical Piece of record.

* Main Memory

→ Memory that is internal to the Computer System is Program that can be loaded into registers for subsequent execution or processing.

* Malicious Software

→ Any software that is designed to cause or harm other people's operating system or to steal any data from them is known as malicious Software.

→ In some cases it spreads itself to the other computers via Email or infected disks types of this includes viruses, trojan horse, worms.



* Memory Cycle time

- The time it takes to read one word, this is the inverse of the rate which words can be read or written to the memory.

* Memory Partition

- The subdivision of storage into independent sections.

* Micro kernel

- A Small Privileged operating System Core that Provides Process, Memory management and kernel management Services to Perform Some of the functions traditionally associated with the operating System Kernel.



* Multi Processing

→ A mode of operation that provides parallel processing by two or more processors.

* Multi Programming

→ A mode of operation that provides for the interleaved execution of two or more computer programs by a single processor the same as multitasking using different terminology.

* Multi Tasking

→ A mode of operation that provides for concurrent execution for two or more interleaved processes the same as multi programming

* Mutual Exclusion

→ Condition in which there is a set of processes only one of which is able to access a given resource to perform a given function at any time.

* Operating System

→ Software that Controls the Execution of Program and that Provides Services Such as Resource Allocation and Output Control and Data Management.

* Page

→ In virtual storage a fixed length block that has a virtual address and that is transferred as a unit between main memory and Secondary memory.

* Page Fault.

- Occurs when the Page containing a referenced word is not Present in the main memory. This causes an interrupt and requires that the Proper Page be brought into main memory.

* Page Frame

- A fixed size Contiguous block of main memory used to hold a Page.

* Paging

- The transfer of Pages between main memory and secondary memory.

* Physical address

→ The absolute location of a unit of data in memory word or byte in memory block of secondary storage.

* Pipes

→ A circular buffer allowing two processes to communicate on the producer consumer model thus it is a fifo written by one process and consumed by another process.

→ In some systems the pipe is generalized to allow any item in the queue for consumption.

* Preemption

→ Reclaiming a process from a process before the process has finished using it.

* Paging

→ The retrieval of pages other than the demanded by a page fault.

→ The hope is that the additional pages will be loaded in the user buffer conserving disk I/O.

* Process

→ A program in execution a process is controlled and scheduled by an operating system.

* Process Control block

→ It is a data structure containing information about the characteristics and the states of the process.

* Process Migration

→ The transfer of a sufficient amount of the state of a process from one machine to another for the process to execute on the target machine.

* Process state

→ All the information that the operating system needs to manage a process and that a processor needs to properly execute the process.

- The Process state includes the contents of the various processor registers such as the Program Counter and data registers.
- it also includes info of uses in operating system such as the priority of the process and whether the process is waiting for any I/O Event or not.
Same as Execution Context.

* Program Counter (PC)

- Instruction address register.

* Processor

- In a Computer a functional unit that Interacts and Executes information.

→ A Processor consists of atleast an instruction Control Unit and an arithmetic unit.

* Programmed I/O

→ The term of I/O in which the Processor issues an I/O Command to an I/O module and then it must wait for the operation to complete before proceeding any further.

* Real time System

→ An operating system that must schedule and manage Real time things.

* Real time tasks

→ A task that is Executed in Connection with Some Process or function or set of Events External to the Computer System and that must meet one or more deadlines to interact correctly within the Environment.

* Registers

→ high speed memory internal to the CPU. Some registers are user visible that is available to the Programmer via the machine instruction set.

→ other registers are used by the CPU for Control Purposes.

* Relative address

→ An address that is calculated as displacement from a base address.

* Remote Procedure call

→ A technique by which two programs on different machines interact using Procedure Call Return Syntax and Semantics. Both Called and Calling Program belong to the if the Partner Program. were running on the same machine.

* Response time

→ In a data system the Elapsed time between the end of transmission of Enquiry message and the beginning of a response messages.



* Round Robin

- > A scheduling algorithm in which the processes are activated in a fixed order that is all process are in circular queue.
- > A process that cannot proceed because it is waiting for an event to occur.

I/O Event

* Scheduling

- > To select jobs or tasks about are to be dispatched in some operating systems other units of work such as input output may also be scheduled.

* Secondary Memory

- Memory located outside the Computer System itself that is it cannot be Processed directly by the Processor.
- it must first be copied into main memory Examples are disks and tapes.

* Segment

- In Virtual memory a block that has a virtual address
- The blocks of a Program may be of unequal length and may be of even dynamically varying length

* Segmentation

→ The division of a Program or application into segments as part of virtual memory scheme.

* Semaphore

→ An integer value used for signaling among processes. Only three actions must be performed on a semaphore all of which can be atomic.

→ Depending on the exact definition of the semaphore, the decrement operation may result in the blocking of the process also called Counting Semaphores and General Semaphores.

* Sequential file

- A file in which records are ordered according to the values of one or more key fields and processed in some sequence from start of the file.

* Session

- A collection of one or more processes that represent a single interaction user application or operating system function.
- All keyboard and mouse input is directed to foreground session and then it is displayed on the screen.

* Shell

- The portion of operating system that interprets interactive user commands

→ it functions as an interface between the user and the operating system

* Stack

- An ordered list in which items are appended to and deleted from the same end of the list known as top.
- This means that the next item is appended on the top of the list. That is the next item appended to the list is put on the top.
- The next item to be removed from the list is the item that has been in the list the shortest time. This method is characterized as last in first out.



* Starvation

- A condition in which a process is indefinitely delayed because other processes are always given preference.

* Strong Semaphore

- A semaphore in which all processes waiting on the same semaphore are queued and will be eventually processed in the same order as they are executed the FIFO operations.

* Swapping

- A process that changes the contents of an area of main storage with the contents of area in secondary memory.

Numericals

Process	Allocation			Max Available		
P ₀	A	B	C	A	B	C
P ₁	0	1	0	7	5	3
P ₂	2	0	0	3	2	2
P ₃	3	0	2	9	0	2
P ₄	2	1	1	2	2	2
	0	0	2	4	3	3

Need table

	A	B	C
P ₀	7	4	3
P ₁	1	2	2
P ₂	6	0	0
P ₃	0	1	1
P ₄	4	3	1

P₀ 743 \leq 332 \leftarrow Condition fail.

P₁ 122 \leq 332 \leftarrow Condition true

$$W = 332 + 200 \\ = 532.$$

P₂ Need \leq work \leftarrow Condition false

P₃ Need \leq work \leftarrow tree

$$615 \leq 532 \\ = 532 + 211$$

$$= 743$$

P₄ Need < Work W = 7h3 + 002

$$431 < 7h3 \quad = 745$$

P₀ Need < Work

$$7h3 < 745 \quad W = 7h5 + 010 \\ = 755$$

P₂ Need < work W = 755 + 302

$$600 < 755 \quad = 1057$$

Safe Sequence = $\langle P_1, P_3, P_4, P_0, P_2 \rangle$

* Fifo

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0,
 1, 7, 0, 1

7	0	1	2	0	3	0	4	2	3	0	3	0
7	7	7	2		2	2	4	4	4			
0	0	0		3	3	3	2	2				
1	1			1	0	0	0	3				

1	2	0	1	7	0	(
0	0		7	7	7	
1)		1	0	0	
3	2		2	2	1	

Page fault = 15

No. of frames = 3

* LRU

7, 0, 1, 2, 0, 3, 0, h, 2, 3, 0, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

7	0	1	2	0	3	0	h	2	3	0	3	0	3	2
7	7	7	2		2		4	h	9	0				
0	0	0			0		0	0	3	3				3
			1	1	3		3	2	2	2				2

2	0	1	7	0	1
1		1			
0		0			
2		7			

No. of frames = 3

Page fault = 12

* Optimal

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1

7	0	1	2	0	3	0	4	2	3	0	3	2	1
7	7	7	2		2		2		2		2		2
0	0	-		0		0	5		0		0		0
	1	1		3		3		3		3		3	

2 0 1 7 0 1
 7
 0
 1

No of frames = 3

Page fault = 9

* FIFO

1, 3, 0, 3, 5, 6, 3

1	3	0	3	5	6	3
1	1	1		5	5	5
3	3			3	6	6
0				0	0	3

Page fault = 6

No of frames = 3



* Optimal

7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 3

7	0	1	2	0	3	0	4	2	3	0	3	2	3
7	7	7	7		3				3				
0	0	0			0				0				
.	1	1			1				4				
	2				2				2				

Page fault = 6

No of frames = 4

D E P A R T M E N T O F C O M P U T E R S C I E N C E
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – 3

ROLL NO : 26

NAME : RUTWIK PATEL

SUBJECT : OPERATING SYSTEMS

NO.	TITLE	PAGE NO.	DATE	SIGN
1	Basic salary of a person is input through the keyboard. His dearness allowance is 40% of basic salary and house rent is 20% of basic salary. Write a program to calculate the gross pay.		10/12/2020	
2	The distance between two cities is input through the keyboard. (in km). Write a program to convert this distance into metres, feet, inches and centimeters and display the results.		10/12/2020	
3	The length and breadth of a rectangle and radius of a circle are entered through the keyboard, calculate the perimeter and area of rectangle and area and circumference of the circle.		10/12/2020	
4	If a five digit number is entered through the keyboard, calculate the sum of its digits.		10/12/2020	
5	The file /etc/passwd contains info about all users. Write a program which would receive the logname during execution, obtain information about it from the file and display the information on screen in some appropriate format.		10/12/2020	
6	The script will receive the filename or filename with its full path, the script should obtain information about this file as given by "ls -l" and display it in proper format.		10/12/2020	
7	If cost price and selling price of an item are entered through the keyboard, write a program to determine whether the seller has made profit or loss. Also determine how much profit/loss is made.		10/12/2020	
8	Check whether the entered no. is odd or even.		10/12/2020	
9	Check whether the entered no. is prime or not.		10/12/2020	
10	Check whether the entered year is a leap year or not.		10/12/2020	
11	The script receives two file names as arguments, the script must check whether the files are same or not, if they are similar then delete the second file		10/12/2020	
12	Write a script which will display whether your friend has logged in or not, if he has logged in then send him some message.		10/12/2020	
13	While executing a shell script, either the logname or uid is supplied at the command prompt, write a shell script to find out at how many terminals has this user logged in.		10/12/2020	

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – 3**

R O L L N O : 26

N A M E : RUTWIK PATEL

S U B J E C T : OPERATING SYSTEMS

NO.	TITLE	PAGE NO.	DATE	SIGN
14	Write a shell script to display the date with the format :- 25th October 2005 is a Tuesday.		10/12/2020	
15	Write a shell script to display the appropriate message like : Good Morning / Good Afternoon / Good Evening		10/12/2020	
16	Write a shell script to display the menu driven interface :- 1) list all files of the current directory 2) print the current directory 3) print the date 4) print the users otherwise display "Invalid Option".		10/12/2020	
17	Create a menu driven calculator which asks for two integers and perform basic arithmetic operations.		10/12/2020	
18	Find the factorial of any number.		10/12/2020	
19	Display the fibonacci series upto some number		10/12/2020	
20	Two numbers are entered through the keyboard, find the power, one number raised to another.		10/12/2020	
21	Write a script which has the functionality similar to head and tail commands.		10/12/2020	
22	Write a script which reports name and size of all files in a directory. whose sizes exceed 1000. The filenames should be printed in the descending order of their sizes. The total no. of files must be reported		10/12/2020	
23	A friend of yours has promised to log in a particular time. You want to contact him as soon as he logs in, write a script which checks after every minute whether the friend has logged in or not. The logname should be supplied at command prompt.		10/12/2020	
24	Print the prime nos. from 1 to 300.		10/12/2020	
25	Program must display all the combinations of 1, 2, and 3.		10/12/2020	
26	Write a script for renaming each file in the directory such that it will have the current shell PID as an extension. The shell script should ensure that the directories do not get renamed.		10/12/2020	
27	A file called wordfile consists of several words. Write a shell script which will receive a list of filenames, the first of which would be wordfile. The shell script should report all occurrences of each word in wordfile in the rest of the files supplied as arguments.		10/12/2020	

**DEPARTMENT OF COMPUTER SCIENCE
ROLLWALA COMPUTER CENTRE
GUJARAT UNIVERSITY
M.C.A. – 3**

ROLL NO : 26

NAME : RUTWIK PATEL

SUBJECT : OPERATING SYSTEMS

NO.	TITLE	PAGE NO.	DATE	SIGN
28	Write a shell script which deletes all the lines containing the word "unix" in the files supplied as arguments to it.		10/12/2020	
29	The word "unix" is present in only some of the files supplied as arguments to the shell script. Your script should search each of these files in turn and stop at the first file that it encounters containing the word unix. The filename should be displayed on the screen.		10/12/2020	
30	A shell script receives even number of filenames. Suppose four filenames are supplied then the first file should get copied into second file, the third file should get copied into fourth and so on.. If odd number of filenames are supplied display error message.		10/12/2020	
31	The script displays a list of all files in the current directory to which you have read, write and execute permissions.		10/12/2020	
32	The script receives any number of filenames as arguments. It should check whether every argument supplied is a file or directory. If it is a directory it should be reported. If it is a filename then name of the file as well as the number of lines present in it should be reported.		10/12/2020	
33	A script will receive any number of filenames as arguments. It should check whether such files already exist. If they do, then it should be reported, if not then check if a subdirectory "mydir" exists or not in the current directory, if it doesn't exist then it should be created and in it the files supplied as arguments should be created.		10/12/2020	
34	Accept the marks of 5 subjects and calculate the percentage and grade. 35. Print Armstrong nos. from 1 to 500.		10/12/2020	
35	Print Armstrong nos. from 1 to 500.		10/12/2020	
36	Accept the measure (angles) of a triangle and display the type of triangle. (eg. acute, right, obtuse)		10/12/2020	
37	Display all the numbers from 1 to 100 which are divisible by 7		10/12/2020	
38	Find the largest and smallest of 3 different numbers.		10/12/2020	
39	Find HCF and LCM of a given no.		10/12/2020	
40	Display the dates falling on Sundays of the current month.		10/12/2020	

```
*****  
*****  
Name: Rutwik Patel  
Class: MCA III  
Roll No.: 26  
Subject: Operating System - OS  
*****  
*****
```

Q1) Basic salary of a person is input through the keyboard. His dearness allowance is 40% of basic salary and house rent is 20% of basic salary. Write a program to calculate the gross pay.

```
*****  
*****  
echo "Enter Salary : "  
read salary  
  
announces=$((salary * 40 / 100))  
echo "Announces is : " $announces  
rent=$((salary * 20 / 100))  
echo "Rent is : " $rent  
gross_pay=`expr $salary + $announces + $rent`  
echo "Gross Pay Salary = " $gross_pay  
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh grosspay.sh  
Enter Salary :  
20000  
Announces is : 8000  
Rent is : 4000  
Gross Pay Salary = 32000  
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$  
*****  
*****  
Q2)
```

The distance between two cities is input through the keyboard(in km). Write a program to convert this distance into metres, feet, inches and centimeters and display the results.

```
*****  
*****  
echo "Enter the distance in kilometers : "  
read km  
meter=`expr $km \* 1000`  
echo " Distance in meter is : " $meter  
  
cm=`expr $meter \* 100`  
echo " Distance in centimeter is : " $cm  
  
feet=`echo " scale = 2; $km * 3280.84" | bc`  
echo " Distance in feet is : " $feet  
  
inches=`echo " scale = 2; $km * 39370.08" | bc`  
*****
```

```
echo " Distance in inches is : " $inches
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh distance.sh  
Enter the distance in kilometers :  
2  
Distance in meter is : 2000  
Distance in centimeter is : 200000  
Distance in feet is : 6561.68  
Distance in inches is : 78740.16  
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
Q3)
```

The length and breadth of a rectangle and radius of a circle are entered through the keyboard, calculate the perimeter and area of rectangle and area and circumference of the circle.

```
*****  
*****
```

#area of rectangle and the circle

```
echo "Enter the length of the rectangle : "  
read len  
echo "Enter he breathe of the rectangle : "  
read breath  
area=`echo "scale = 2; $len * $breath " | bc`  
echo "area of the rectangle is : " $area  
peri=`echo "scale = 2; $area * 2 " | bc`  
echo "perimeter of the rectangle is :" $peri  
echo "Enter the radius of the circle : "  
read r  
a=`echo " scale = 2 ; 3.14 * $r * $r " | bc`  
echo "The area of the circle is : " $a  
cir=`echo " scale = 2 ; 2 * 3.14 * $r " | bc`  
echo "The circumference of the circle is : " $cir
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh area.sh  
Enter the length of the rectangle :  
10  
Enter he breathe of the rectangle :  
2.5  
area of the rectangle is : 25.0  
perimeter of the rectangle is : 50.0  
Enter the radius of the circle :  
5.5  
The area of the circle is : 94.98  
The circumference of the circle is : 34.54
```

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
*****  
Q4)
```

If a five digit number is entered through the keyboard, calculate the sum of its digits.

```
*****  
*****
```

```
# Sum of all the digit of the number
```

```
echo "Enter a five digit number : "
read num
number=$num
sum=0
while [ "$num" -gt 0 ]
do
    r=$((num % 10))
    sum=`expr $sum + $r`
    num=`expr $num / 10`
done
echo " The sum of " $number " is : " $sum
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh sum.sh
Enter a five digit number :
12345
The sum of 12345 is : 15
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
*****  
Q5)
```

The file /etc/passwd contains info about all users. Write a program which would receive the logname during execution, obtain

information about it from the file and display the information on screen in some appropriate format. (Hint : use cut)

eg. Logname : UID : GID : Default working directory : Default working shell

```
*****  
*****
```

```
cut -f1,3,4,6,7 -d":" /etc/passwd | tail -n 1
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_5.sh
Rutwik:1000:1000:/home/Rutwik:/bin/bash
```

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
*****  
Q6)
```

The script will receive the filename or filename with its full path, the script should obtain information about this file as given by "ls -l" and display it in proper format.
eg. Filename : File access permissions : Number of links : Owner of the file : Group to which belongs : Size of file : File modification date : File modification time

```
*****  
*****
```

```
echo "Enter file name :"  
read file
```

```
ls -l $file | cut -d' ' -f 1,2,3,4,5,6,7,8,9 -s | awk '{print $9 ":" $1  
":" $2 ":" $3 ":" $4 ":" $5 ":" $6 ":" $7 ":" $8}'
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_6.sh  
Enter file name :  
sum.sh  
sum.sh:-rwxrwxrwx:1:Rutwik:Rutwik:239:Nov:21:10:33  
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
*****  
Q7)
```

If cost price and selling price of an item are entered through the keyboard, write a program to determine whether the seller has made profit or loss. Also determine how much profit/loss is made.

```
*****  
*****
```

```
echo "Enter the cost price of an item :"  
read cost_price  
echo "Enter the selling price of an item :"  
read sell_price  
if [ $sell_price -gt $cost_price ]  
then  
    echo "The seller made profit and that is : " `expr $sell_price -  
$cost_price`  
else  
    echo "The seller made loss and that is : " `expr $cost_price -  
$sell_price`  
fi
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_7.sh
Enter the cost price of an item :
10000
Enter the selling price of an item :
12000
The seller made profit and that is : 2000
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_7.sh
Enter the cost price of an item :
10000
Enter the selling price of an item :
8000
The seller made loss and that is : 2000
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****
Q8)
```

Check whether the entered no. is odd or even.

```
*****
*****
# check the given no is odd or not

echo "Enter the number : "
read num

rem=$((num % 2))
if [ $rem -eq 0 ]
then
        echo "The number is even "
else
        echo "The number is odd "
fi
```

```
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_8.sh
Enter the number :
12
The number is even
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****
```

Q9)

Check whether the entered no. is prime or not.

```
*****
*****
```

check the no is prime or not

```
echo "Enter the number you want to check the prime or not : "
read num
```

```

i=2
flag=0

while test $i -le `expr $num / 2`
do
    if test `expr $num % $i` -eq 0
    then
        flag=1
    fi
    i=`expr $i + 1`
done

if [ $num -eq 1 ]
then
    echo $num " is neither prime nor composite "
else
    if [ $flag -eq 0 ]
    then
        echo $num " is prime number "
    else
        echo $num " is not prime number "
    fi
fi
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_9.sh
Enter the number you want to check the prime or not :
```

```
5
5 is prime number
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_9.sh
Enter the number you want to check the prime or not :
6
6 is not prime number
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****
```

Q10)

Check whether the entered year is a leap year or not.

```
*****
*****
```

```
echo "Enter the year : "
read year
```

```
if [ `expr $year % 4` -eq 0 -a `expr $year % 100` -ne 0 -o `expr $year %
400` -eq 0 ]
then
    echo " Year $year is leap year : "
else
    echo " Year $year is not leap year : "
fi
```

```
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_10.sh
Enter the year :
2020
Year 2020 is leap year :
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_10.sh
Enter the year :
2019
Year 2019 is not leap year :
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****
```

Q11)

The script receives two file names as arguments, the script must check whether the files are same or not, if they are similar then delete the second file.

```
*****  
*****
```

```
# the script must check whether the files are same or not,
# if they are similar then delete the second file.
```

```
#!/bin/bash

if [ ! -f $1 ]; then
    echo "$1 not found!"
    exit
fi
if [ ! -f $2 ]; then
    echo "$2 not found!"
    exit
fi

my_var=$(cmp -b $1 $2)
if test -z "$my_var"
then
    echo "Files are same"
    rm $2
    echo $2 "Deleted"
else
    echo "Files are not same"
fi
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_11.sh
new.txt new1.txt
Files are same
new1.txt Deleted
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****  
Q12)
```

Write a script which will display whether your friend has logged in or not, if he has logged in then send him some message.

```
*****  
*****  
echo "Enter user name "  
read name  
  
who | grep $name > /dev/null  
  
if [ $? -eq 0 ]  
then  
    echo "User is logged in "  
    echo "Please enter a message to send him "  
    read msg  
    echo $msg  
else  
    echo "User is not logged in "  
fi
```

```
*****  
*****
```

Output:

```
Rutwik@Rutwik-VirtualBox:~/Rutwik$ sh s_s_12.sh  
Enter user name  
Rutwik  
User is logged in  
Please enter a message to send him  
Hii !! How are you ?  
Hii !! How are you ?  
Rutwik@Rutwik-VirtualBox:~/Rutwik$ sh s_S_12.sh  
Enter user name  
Piyu  
User is not logged in  
Rutwik@Rutwik-VirtualBox:~/Rutwik$
```

```
*****  
*****  
Q13)
```

While executing a shell script, either the logname or uid is supplied at the command prompt, write a shell script to find out at how many terminals has this user logged in.

```
*****  
*****
```

```
*****  
*****
```

Output

```
*****
*****
Q14)
Write a shell script to display the date with the format :- 25th October
2005 is a Tuesday.
*****
*****
```

9th December 2020 is a Wednesday.
d=`date +%d\ %B\ %Y\ is\ a\ %A`
echo \$d

```
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_14.sh
10 December 2020 is a Thursday
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****
Q15)
Write a shell script to display the appropriate message like : Good
Morning / Good Afternoon / Good Evening
*****
*****
```

```
#get the current hour in (24 hrs format)
hour=$(date +"%H")

if [ $hour -ge 0 -a $hour -lt 12 ]
then
    echo " Good Morning , $USER "
elif [ $hour -ge 12 -a $hour -lt 17 ]
then
    echo " Good Afternoon , $USER "
elif [ $hour -ge 17 -a $hour -lt 19 ]
then
    echo " Good Evening , $USER "
else
    echo "Good Night , $USER "
fi
```

```
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_15.sh
Good Night , Rutwik
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$
```

```
*****
*****
Q16)
Write a shell script to display the menu driven interface :- 1) list all
files of the current directory, 2) print the current
```

```

directory, 3) print the date, 4) print the users otherwise display
"Invalid Option".
*****
*****
echo "Press 1 for list all files of the current directory "
echo "Press 2 for print the current directory "
echo "Press 3 for print the date "

ch=1

while [ ch -ne 0 ]
do

    echo "Enter your choice from the above : "
    read ch

    case $ch in
    1)
        ls -l
        ;;
    2)
        pwd
        ;;
    3)
        date
        ;;
    *)
        echo "Invalid date : "
        ;;
    ;;
done

*****
*****

```

Output:

```

Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_16.sh
Press 1 for list all files of the current directory
Press 2 for print the current directory
Press 3 for print the date
Enter your choice from the above :
1
total 0
-rwxrwxrwx 1 Rutwik Rutwik 550 Nov 21 12:05 area.sh
-rwxrwxrwx 1 Rutwik Rutwik 360 Nov 21 10:31 distance.sh
-rwxrwxrwx 1 Rutwik Rutwik 243 Nov 21 10:29 grosspay.sh
-rwxrwxrwx 1 Rutwik Rutwik 214 Nov 21 21:05 s_s_10.sh
-rwxrwxrwx 1 Rutwik Rutwik 227 Nov 21 21:58 s_s_12.sh
-rwxrwxrwx 1 Rutwik Rutwik 325 Nov 23 19:56 s_s_15.sh
-rwxrwxrwx 1 Rutwik Rutwik 391 Nov 23 20:17 s_s_16.sh
-rwxrwxrwx 1 Rutwik Rutwik 299 Nov 22 17:22 s_s_18.sh
-rwxrwxrwx 1 Rutwik Rutwik 23 Nov 22 17:24 s_s_19.sh
-rwxrwxrwx 1 Rutwik Rutwik 555 Nov 23 19:46 s_s_34.sh
-rwxrwxrwx 1 Rutwik Rutwik 46 Nov 21 12:19 s_s_5.sh
-rwxrwxrwx 1 Rutwik Rutwik 161 Nov 21 19:16 s_s_6.sh
-rwxrwxrwx 1 Rutwik Rutwik 326 Nov 21 19:34 s_s_7.sh
-rwxrwxrwx 1 Rutwik Rutwik 183 Nov 21 18:23 s_s_8.sh
-rwxrwxrwx 1 Rutwik Rutwik 422 Nov 21 20:37 s_s_9.sh

```

```
-rwxrwxrwx 1 Rutwik Rutwik 246 Nov 21 19:11 sum.sh
Enter your choice from the above :
2
/mnt/e/Rolwala/sem-3/o.s/practices
Enter your choice from the above :
3
Mon Nov 23 20:18:02 IST 2020
Enter your choice from the above :
0
Invalid Option :
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$
```

```
*****  
*****  
Q17)  
Create a menu driven calculator which asks for two integers and perform  
basic arithmetic operations.  
*****  
*****  
*****  
echo "Enter 1st Integer :"  
read no1  
echo "Enter 2nd Integer :"  
read no2  
ch=1  
while [ $ch -ne 0 ]  
do  
    echo  
"\n1.Addition\n2.Subtraction\n3.Multiplication\n4.Division\n0.Exit\n"  
    echo "Enter Your Choice :"  
    read ch  
  
    case $ch in  
        "1") ans=`expr $no1 + $no2`  
        echo "Addition = " $ans ;;  
  
        "2") ans=`expr $no1 - $no2`  
        echo "Subtraction = " $ans ;;  
  
        "3") ans=`expr $no1 \* $no2`  
        echo "Multiplication = " $ans ;;  
  
        "4") ans=`expr $no1 \/\ $no2`  
        echo "Division = " $ans ;;  
  
        "0") echo "Exit..";;  
  
        *) echo "Invalid Choice.."  
  
    esac  
done
```

```
*****  
*****  
Output:  
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_17.sh  
Enter 1st Integer :
```

```
20
Enter 2nd Integer :
10
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit
```

```
Enter Your Choice :
1
Addition = 30
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit
```

```
Enter Your Choice :
2
Subtraction = 10
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit
```

```
Enter Your Choice :
3
Multiplication = 200
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit
```

```
Enter Your Choice :
4
Division = 2
```

```
1.Addition
2.Subtraction
3.Multiplication
4.Division
0.Exit
```

```
Enter Your Choice :
0
Exit..
```

```
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$
```

```
*****
*****
Q18)
Find the factorial of any number.
*****
```

```
*****
#shell scr:wqipt for factorial of a number
#factorial using while loop

echo "Enter a number"
read num

fact=1

while [ $num -gt 1 ]
do
    fact=$((fact * $num)) #fact = fact * num
    num=$((num - 1))      #num = num - 1
done
echo "FActorial of the number " $num " is " $fact

*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_18.sh
Enter a number
5
FActorial of the number 5 is 120
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****  
Q19)
Display the fibonacci series upto some number.
*****
```

```
#Fibonnaci series :
echo "How many terms you want to generated "
read n
```

```
x=0
y=1
i=2
echo "Fibonacci series upto $n numbers :"
echo "$x"
echo "$y"

while [ $i -lt $n ]
do
    i=`expr $i + 1 `
    z=`expr $x + $y `
    echo "$z"
    x=$y
    y=$z
done
```

```
*****
*****
```

Output:

```

Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_19.sh
How many terms you want to generated
5
Fibonacci series upto 5 numbers :
0
1
1
2
3
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ ****
***** Q20)
Two numbers are entered through the keyboard, find the power, one number
raised to another.
***** echo "Enter Exponent : "
read exp
echo "Enter Power : "
read pow

i=0
ans=1

while [ $i -lt $pow ]
do
    ans=`expr $ans \* $exp`
    i=`expr $i + 1`
done

echo "$exp ^ $pow = $ans"
***** Output:
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_20.sh
Enter Exponent :
5
Enter Power :
2
5 ^ 2 = 25
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ sh s_s_20.sh
Enter Exponent :
2
Enter Power :
5
2 ^ 5 = 32
Rutwik@RutwikPatel:/mnt/e/Rolwala/sem-3/o.s/practices$ ****
***** Q22)

```

Write a script which reports name and size of all files in a directory. whose sizes exceed 1000. The filenames should be printed in the descending order of their sizes. The total no. of files must be reported.

```
# all files in a directory whose sizes exceed 1000.  
# The filenames should be printed in the descending order of their  
sizes.  
# The total no. of files must be reported.
```

```
#! /bin/bash  
ls --sort=size -l | awk '$5 >= 500 {print $5,$9}'
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_22.sh  
151552 which  
4096 mydir  
1994 s_s_17.sh  
1650 s_s_16.sh  
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

Q24)

Print the prime nos. from 1 to 300.

```
fi  
  
if [ $n -le 3 ]  
then  
    return 1  
fi  
  
if [ $($n%2) -eq 0 ]  
then  
    return 0  
fi  
  
if [ $($n%3) -eq 0 ]  
then  
    return 0  
fi  
i=5  
  
while [ $($i*$i)) -eq 0 ]  
do  
    if [ $($n%$i)) -eq 0 ]  
    then  
        return 0
```

```

        fi
        if [ $($n%($i+2)) -eq 0 ]
        then
            return 0
        fi
        i=$((i+6))
    done
    return 1
}

num=2

while [ $num -le 300 ]
do
    checkprime $num
    isprime=$?

    if [ $isprime -eq 1 ]
    then
        echo "$num"
    fi

    num=$((num+1))
done

```

Output :

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_24.sh
2
3
5
7
11
13
17
19
23
25
29
31
35
37
41
43
47
49
53
55
59
61
65
67
71
73
77
79
83
85
89
91

```

95
97
101
103
107
109
113
115
119
121
125
127
131
133
137
139
143
145
149
151
155
157
161
163
167
169
173
175
179
181
185
187
191
193
197
199
203
205
209
211
215
217
221
223
227
229
233
235
239
241
245
247
251
253
257
259
263
265
269
271

```
275  
277  
281  
283  
287  
289  
293  
295  
299
```

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****
```

```
*****  
*****
```

Output:

```
*****  
*****
```

```
*****  
*****
```

```
Q25)
```

```
Program must display all the combinations of 1, 2, and 3.
```

```
*****  
*****
```

```
*****  
*****
```

```
for i in 1 2 3  
do  
    for j in 1 2 3  
    do  
        for k in 1 2 3  
        do  
            if [ $k -le $i ]  
            then  
                if [ $j -le $i ]  
                then  
                    echo $i $j $k  
                fi  
            fi  
        done  
    done  
done
```

```
*****  
*****
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_25.sh
```

```
1 1 1  
2 1 1  
2 1 2  
2 2 1  
2 2 2  
3 1 1  
3 1 2  
3 1 3  
3 2 1  
3 2 2
```

```

3 2 3
3 3 1
3 3 2
3 3 3
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ ****
***** Q26)
Write a script for renaming each file in the directory such that it will
have
the current shell PID as an extension. The shell script should ensure
that
the directories do not      get renamed.
***** for f in *
do
    [ -e $f ] || continue
    mv $f $f.$$
done
*****

```

Output:

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices/mydir$ ls -l
total 0
-rwxrwxrwx 1 Rutwik Rutwik 13 Dec  9 16:38 Hello.sh.82
-rwxrwxrwx 1 Rutwik Rutwik 58 Dec  9 16:38 file1.txt.82
-rwxrwxrwx 1 Rutwik Rutwik  0 Dec  9 14:42 Rutwik.sh.82
-rwxrwxrwx 1 Rutwik Rutwik  0 Dec  9 16:43 s_s_26.sh
-rwxrwxrwx 1 Rutwik Rutwik 55 Dec  9 16:39 s_s_26.sh.82
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices/mydir$ ****
*****
```

Q27)

A file called wordfile consists of several words. Write a shell script which will receive a list of filenames, the first of which would be wordfile. The shell script should report all occurrences of each word in wordfile in the rest of the files supplied as arguments.

```
*****
```

```

# file called wordfile consists of several words.
# Write a shell script which will receive a list of filenames,
# the first of which would be wordfile.
# The shell script should report all occurrences of each word
# in wordfile in the rest of the files supplied as arguments

#!/bin/bash

if [ $# -eq 0 ]; then
    printf "Usage:\n"
    echo "./27-findWordFromFile.sh <wordFile> <findFile ...>"
```

```

        exit
fi

filesToRead=$(( $#-1 ))
echo $filesToRead

# Reading Line by Line
while read line; do
# Reading Word by Word
    for word in $line; do
        echo "Searching word: '$word' ..."
        # 2 is slice starting index
        # filesToRead is slice length
        grep --color=always -n $word ${@:2:$filesToRead}
        printf "Done.\n\n"
    done
done <"$1" # $1 is the file name we want to search
*****
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_27.sh
Usage:
./27-findWordFromFile.sh <wordFile> <findFile ...>
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****
*****
```

Q28)
Write a shell script which deletes all the lines containing the word
"unix"
in the files supplied as arguments to it

```
*****
##!/bin/bash
#
=====
# sed '/foo/d' deleteFromFile.txt
#
# the substring foo inside the foobar string is also replaced.
# If this is not the wanted behavior, use the word-boundary expression
# (\b)
# at both ends of the search string.
# This ensures the partial words are not matched.
#
=====
word="UNIX"
#
# Read all args
for i
do
    # I is for Insensitive
    # d is for delete
```

```

# I must be written first
sed -i "/\b$word\b/Id" $1
done

*****
*****#Output:

#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_28.sh
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_28.sh
unix.txt
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat unix.txt
#hello
#nee
l@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi unix.txt
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat unix.txt
#hello
#unix is the best
#i love linux
#unix is my favourite os
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_28.sh
unix.txt
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat unix.txt
#hello
#i love linux
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

Q29)

The word "unix" is present in only some of the files supplied as arguments to the shell script. Your script should search each of these files in turn and stop at the first file that it encounters containing the word unix. The filename should be displayed on the screen.

```

for i
do
    echo "Searching file : $i..."

    if grep -q "unix" "$i";then
        echo "Found in $i"
        exit
    fi
    echo "done"
done
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_29.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_29.sh
file.txt
```

```

Searching file : file.txt...
Found in file.txt
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ ****
****

Q30)
A shell script receives even number of filenames. Suppose four filenames
are
supplied then the first file should get copied into second file,
the
third file should get copied into fourth and so on.. If odd number of
filenames are supplied display error message
****

*****
#zero arguments

if [ $# -eq 0 ]
then
    echo "No arguments"
    exit
fi

prevFile=$1

#if even no of args

if [ $(echo $# % 2 | bc) -eq 0 ]
then
    #looping through each argument
    count=1
    for i
    do
        if !((count%2))
        then
            cp $prevFile $i
            echo "'$prevFile' copied to -> $i"
        else
            prevFile=$i
        fi
        count=$(echo $count+1 | bc)
    done
#else odd no of args
else
    echo "Odd no of arguments"
    exit
fi

*****
*****
```

Output:

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ touch new1.txt
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat new1.txt
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi new.txt
```

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat new.txt
Hello
My name is Rutwik
class MCA-3
Have a good day
Nice to meet you
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_30.sh
new.txt new1.txt
s_s_30.sh: 19: [: !1: unexpected operator
s_s_30.sh: 19: [: !2: unexpected operator
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_30.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_30.sh
new.txt new1.txt
s_s_30.sh: 19: Syntax error: "(" unexpected (expecting "then")
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_30.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_30.sh
new.txt new1.txt
s_s_30.sh: 19: Syntax error: "(" unexpected (expecting "then")
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_30.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_30.sh
new.txt new1.txt
s_s_30.sh: 19: 1%2: not found
cp: 'new.txt' and 'new.txt' are the same file
'new.txt' copied to -> new.txt
s_s_30.sh: 19: 2%2: not found
'new.txt' copied to -> new1.txt
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat new.txt
Hello
My name is Rutwik
class MCA-3
Have a good day
Nice to meet you
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ cat new1.txt
Hello
My name is Rutwik
class MCA-3
Have a good day
Nice to meet you
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_S_30.sh
*****
*****
```

Q31)

The script displays a list of all files in the current directory to which you have read, write and execute permissions.

```

echo "The list of File Names in the curent directory."
echo "Which have Read,Write and Execute permisions. "
for file in *
do
    if [ -f $file ]
    then
        if [ -r $file -a -w $file -a -x $file ]
        then
            ls -l $file
        fi
    fi
```

```
done
```

```
*****  
*****  
Output:
```

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_S_31.sh  
The list of File Names in the current directory.  
Which have Read,Write and Execute permissions.  
-rwxrwxrwx 1 Rutwik Rutwik 550 Nov 21 12:05 area.sh  
-rwxrwxrwx 1 Rutwik Rutwik 360 Nov 21 10:31 distance.sh  
-rwxrwxrwx 1 Rutwik Rutwik 243 Nov 21 10:29 grosspay.sh  
-rwxrwxrwx 1 Rutwik Rutwik 227 Nov 23 22:09 s_S_31.sh  
-rwxrwxrwx 1 Rutwik Rutwik 214 Nov 21 21:05 s_s_10.sh  
-rwxrwxrwx 1 Rutwik Rutwik 227 Nov 21 21:58 s_s_12.sh  
-rwxrwxrwx 1 Rutwik Rutwik 325 Nov 23 19:56 s_s_15.sh  
-rwxrwxrwx 1 Rutwik Rutwik 391 Nov 23 20:17 s_s_16.sh  
-rwxrwxrwx 1 Rutwik Rutwik 670 Nov 23 20:39 s_s_17.sh  
-rwxrwxrwx 1 Rutwik Rutwik 299 Nov 22 17:22 s_s_18.sh  
-rwxrwxrwx 1 Rutwik Rutwik 246 Nov 23 21:17 s_s_19.sh  
-rwxrwxrwx 1 Rutwik Rutwik 181 Nov 23 21:41 s_s_20.sh  
-rwxrwxrwx 1 Rutwik Rutwik 235 Nov 23 21:57 s_s_24.sh  
-rwxrwxrwx 1 Rutwik Rutwik 555 Nov 23 19:46 s_s_34.sh  
-rwxrwxrwx 1 Rutwik Rutwik 46 Nov 21 12:19 s_s_5.sh  
-rwxrwxrwx 1 Rutwik Rutwik 161 Nov 21 19:16 s_s_6.sh  
-rwxrwxrwx 1 Rutwik Rutwik 326 Nov 21 19:34 s_s_7.sh  
-rwxrwxrwx 1 Rutwik Rutwik 183 Nov 21 18:23 s_s_8.sh  
-rwxrwxrwx 1 Rutwik Rutwik 422 Nov 21 20:37 s_s_9.sh  
-rwxrwxrwx 1 Rutwik Rutwik 246 Nov 21 19:11 sum.sh  
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$  
*****
```

```
*****  
*****
```

```
Q32)
```

```
The script receives any number of filenames as arguments. It should check  
whether every argument supplied is a file or directory. If it is a  
directory it should be reported. If it is a filename then name of  
the  
file as well as the number of lines present in it should be reported.  
*****
```

```
*****  
*****
```

```
for i; do  
    if [ -d $i ]; then  
        echo "$i -> directory"  
    elif [ -f $i ]; then  
        printf "$i -> file with lines: "  
        wc -l $i | awk {'print $1'}  
    else  
        echo "$i -> Invalid"  
    fi  
done
```

```
*****  
*****
```

```
#output
```

```

#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practiclis$ sh s_s_32.sh
file.txt
#file.txt -> file with lines: 2
#Rutwik@NeIiana:/mnt/e/ROLWALA/sem-3/o.s/practices$



*****



*****Q33)
A script will receive any number of filenames as arguments. It should
check
whether such files already exist. If they do, then it should be
reported, if not then check if a subdirectory "mydir" exists or
not
in the current directory, if it doesnt exist then it should be
created and in it the files supplied as arguments should be created.
*****



*****



# script will receive any number of filenames as arguments.
# It should check whether such files already exist.
# If they do, then it should be reported,
# if not then check if a subdirectory "mydir" exists
# or not in the current directory,
# if it doesnt exist then it should be created and in it
# the files supplied as arguments should be created.

#!/bin/bash

if [ $# -eq 0 ]; then
    echo "No Arguments passed"
    exit
fi

for i;
do
    # If file exists
    if [ -f $i ]; then
        echo "$i exists"
    else
        # if "mkdir" exists
        if [ -d "mydir" ]; then
            # Directory exists
            printf ""
        else
            mkdir mydir
        fi
        touch mydir/$i
        echo "$i file created in \"mydir\""
    fi
done

*****



*****



Output:

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_33.sh
new.txt

```

```
new.txt exists
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****
```

Q34)

Accept the marks of 5 subjects and calculate the percentage and grade.

```
*****  
*****
```

```
#calculate the percentage and marks of 5 subjects
```

```
echo "Enter the five subject marks for the student"
read m1 m2 m3 m4 m5
sum1=`expr $m1 + $m2 + $m3 + $m4 + $m5`
echo "Sum of 5 subjects are: " $sum1
per=`expr $sum1 / 5`
echo "The percentage of the marks of five subject is : " $per
```

```
if [ $per -ge 80 ]
then
    echo " you got Grade : A "
elif [ $per -ge 70 ]
then
    echo " You got Grade : B "
elif [ $per -ge 60 ]
then
    echo " You got grade : C "
elif [ $per -ge 50 ]
then
    echo "You got Grade : D "
else
    echo "FAIL :"
fi
```

```
*****  
*****
```

Output:

```
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_34.sh
Enter the five subject marks for the student
50 80 70 90 60
Sum of 5 subjects are: 350
The percentage of the marks of five subject is : 70
You got Grade : B
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```
*****  
*****
```

Q35)

Print armstrog nos. from 1 to 500.

```
*****  
*****
```

```
*****  
*****
```

```
i=1
while [ $i -lt 500 ]
do
    j=$i
```

```

total=0
while [ $j -gt 0 ]
do
    temp=$(echo $j%10 | bc)
    sum=$(echo ${temp}^3 | bc)
    total=$(echo $total+$sum | bc)
    j=$(echo $j/10 | bc)
done
if [ $total -eq $i ]
then
    echo "Armstrong number : " $i
fi
i=`expr $i + 1`
done
*****
*****
Output:

```

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_35.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_35.sh
Armstrong number : 1
Armstrong number : 153
Armstrong number : 370
Armstrong number : 371
Armstrong number : 407
*****
```

Q36)

Accept the measure (angles) of a triangle and display the type of triangle.
 (eg. acute, right, obtuse)

```

echo "Enter angle "
read angle

if [ $angle -ge 0 -a $angle -lt 90 ]
then
    echo "Acute angle"
elif [ $angle -eq 90 ]
then
    echo "Right angle "
elif [ $angle -gt 90 -a $angle -le 180 ]
then
    echo "Obtuse angle "
else
    echo "Incorrect input"
fi
*****
```

```

*****
*****
Output:

```

```

Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_36.sh
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_36.sh

```

```

Enter angle
120
Obtuse angle
Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ ****
***** Q37)
Display all the numbers from 1 to 100 which are divisible by 7.
***** ****
***** checkDivisible(){
    n=$1
    if [ $((n % 7)) -eq 0 ]; then
        return 1
    fi
    return 0
}
num=1
echo "The following no is devisible by 7 \n"
while [ $num -le 100 ]
do
    checkDivisible $num
    isDivisible=$?

    if [ $isDivisible -eq 1 ]
    then
        printf "$num "
    fi
    num=$((num+1))
done
echo "\n"
***** ****
***** Output:
#output
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ vi s_s_37.sh
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_37.sh
#The following no is devisible by 7
#7 14 21 28 35 42 49 56 63 70 77 84 91 98
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ ****
***** ****

```

Q38)

Find the largest and smallest of 3 different numbers.

```
*****  
*****  
echo "Enter 1st Number :"  
read no1  
echo "Enter 2nd Number :"  
read no2  
echo "Enter 3rd Number :"  
read no3  
  
if [ $no1 -gt $no2 -a $no1 -gt $no3 ]  
then  
    echo "$no1 is Largest.."  
elif [ $no2 -gt $no1 -a $no2 -gt $no3 ]  
then  
    echo "$no2 is Largest.."  
else  
    echo "$no3 is Largest.."  
fi  
  
if [ $no1 -lt $no2 -a $no1 -lt $no3 ]  
then  
    echo "$no1 is Smallest.."  
elif [ $no2 -lt $no1 -a $no2 -lt $no3 ]  
then  
    echo "$no2 is Smallest.."  
else  
    echo "$no3 is Smallest.."  
fi
```

```
*****  
*****  
Output:
```

#output

```
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_38.sh  
#Enter 1st Number :  
#3  
#Enter 2nd Number :  
#6  
#Enter 3rd Number :  
#1  
#6 is Largest..  
#1 is Smallest..  
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

Q39)

Find HCF and LCM of a given no.

```
*****  
*****  
*****
```

```

echo -n "Enter first number : "
read num1
echo -n "Enter second number : "
read num2

max=$num1
den=$num2

if [ $num2 -gt $max ]
then
    max=$num2
    den=$num1
fi

rem=$((max % den))

while [ $rem -ne 0 ]
do
    max=$den
    den=$rem
    rem=$((max % den))
    max=$((max - 1))
done

gcd=$den
lcm=`expr $num1 \* $num2 / $gcd` 

echo "HCF of $num1 and $num2 = $gcd"
echo "LCM of $num1 and $num2 = $lcm"

*****
*****
Output:

```

```

#output

#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_39.sh
#Enter first number : 15
#Enter second number : 55
#HCF of 15 and 55 = 5
#LCM of 15 and 55 = 165
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$
```

```

*****
*****
Q40)
Display the dates falling on Sundays of the current month.
*****
*****o "Sundays in current month are:"
```

```

echo " ----- Using AWK ----- "
cal | awk 'FNR > 2{print $1}'
```

```
*****
```

```
*****
```

Output:

#output

```
#Rutwik@RutwikPatel:/mnt/e/ROLWALA/sem-3/o.s/practices$ sh s_s_40.sh
#s_s_40.sh: 1: o: not found
# ----- Using AWK -----
# 1
# 6
# 13
# 20
# 27
```

```
*****
```

```
*****
```