A close-up of a sign

AI-generated content may be incorrect.

**TYBCA SEM-5**

**501 : Linux Operating System (LOS)**

**Practical Journal**

|  |  |  |  |
| --- | --- | --- | --- |
| **INDEX** | | | |
| **NO** | **PROGRAMS** | **PAGE NO** | **SIGN** |
| **1.** | Write script using case statement to perform basic math operations  ( + , - , \* , / , % ). |  |  |
| **2.** | Write a shell script to reverse a given number. |  |  |
| **3.** | Write script to check inputted file is regular file , directory or does not exist . |  |  |
| **4.** | Write a script which enters username s& password & check that if the username = sugc& password=98765 then display the valid user message. Otherwise invalid user. [script gives maximum 3 attempts to the user.] |  |  |
| **5.** | Accept a string from terminal and echo suitable message if it does not have at least 10 characters. |  |  |
| **6.** | Write a script to delete all vowels from particular string. |  |  |
| **7.** | Write a script to accept a number from user until he enters 0 & find sum of all that numbers. |  |  |
| **8.** | Write a script to Check whether file is empty or not, And if file name doesn‟t exist than print appropriate message. |  |  |
| **9.** | Write a shell script to prompts the user to enter the time (in 24-hour format) then wish them “Good morning”, “Good afternoon”, “Good evening”, or  “Good night” based on the input. (example: If Input is 13 then print Good afternoon) |  |  |
| **10.** | Write a shell script which takes input of file name and prints first 10 lines of that file. file name is to be passed as command line argument. If argument is not passed then any „C‟ program from the current directory is to be selected. |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| **NO** | **WRITE COMMAND USING sed OR grep** | **PAGE NO** | **SIGN** |
| **1.** | Write a command to replace „RAM‟ with „ROM‟ on line no 10 to 20. |  |  |
| **2.** | Display all blank lines between line 20 and 30 of file test.txt. |  |  |
| **3.** | To list file names consist of only 4 digits. |  |  |
| **4.** | Display the lines that do not contain “Unix”. |  |  |
| **5.** | Display the lines which are starting with 1 at the beginning. |  |  |
| **6.** | Display lines beginning either with alphabet or digit from file test.txt. |  |  |
| **7.** | Write a command to display all file name containing only digits in a filename. |  |  |
| **8.** | Display two lines starting from 4th line of file test.txt. |  |  |
| **9.** | To display lines beginning with numbers of a file y1.txt. |  |  |
| **10.** | To count number of words in line 10 thought 20 of file test.txt. |  |  |
|  | | | |
| **WRITE THE COMMAND** | | | |
| **1.** | Count number of characters in first five lines of file x1. |  |  |
| **2.** | Display files of current directory whose 1st character is not digit. |  |  |
| **3.** | Display last 2 lines of working directory. |  |  |
| **4.** | Display only those files of current directory which is own by the current user. |  |  |
| **5.** | To combine content of two file do not use cat command. |  |  |
| **6.** | Count the total no of blank lines of file x1. |  |  |
| **7.** | Display the lines which are not starting with 2 at the beginning. |  |  |
| **8.** | Count the total no. of lines in a file. |  |  |
| **9.** | To display lines beginning with alphabets of a file test.txt |  |  |
| **10.** | Display lines of file from line 3-5. |  |  |

**Programs**

1. **Write script using case statement to perform basic math operations (+, -, \*, /, %).**

echo "Enter First Number:"

read a

echo "Enter Second Number:"

read b

echo "Enter Operation (+, -, \*, /, %):"

read op

case $op in

+) result=$((a + b))

echo "Result = $result"

;;

-) result=$((a - b))

echo "Result = $result"

;;

\\*) result=$((a \* b))

echo "Result = $result"

;;

/) if [ $b -ne 0 ]; then

result=$((a / b))

echo "Result = $result"

else

echo "Error: Division by zero not allowed"

fi

;;

%) if [ $b -ne 0 ]; then

result=$((a % b))

echo "Result = $result"

else

echo "Error: Modulo by zero not allowed"

fi

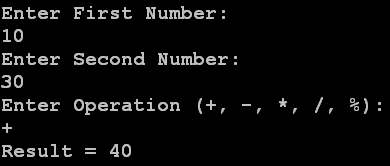
;;

\*) echo "Invalid operation"

;;

esac

**Out Put:**

****

**A black screen with white text

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**

1. **Write a shell script to reverse a given number.**

echo "Enter a Number:"

read num

rev=0

while [ $num -gt 0 ]

do

rem=$((num % 10))

rev=$((rev \* 10 + rem))

num=$((num / 10))

done

echo "Reversed Number = $rev"

**Out Put:**

**A black background with white text

AI-generated content may be incorrect.**

1. **Write script to check inputted file is regular file, directory or does not exits.**

echo "Enter file or directory name:"

read name

if [ -f "$name" ]; then

echo "$name is a regular file."

elif [ -d "$name" ]; then

echo "$name is a directory."

else

echo "$name does not exist."

Fi

**Out Put:**

**A black background with white text

AI-generated content may be incorrect.**

1. **Write a script which enters username & password & check that if the username = sugc & password = 98765 then display the valid user message. Otherwise invalid user. [script gives maximum 3 attempts to the user.]**

username="chintan"

password="12345678"

attempt=1

max\_attempts=3

while [ $attempt -le $max\_attempts ]

do

echo "Attempt $attempt of $max\_attempts"

echo -n "Enter Username: "

read user

echo -n "Enter Password: "

read pass

if [ "$user" = "$username" ] && [ "$pass" = "$password" ];

then

echo "Valid User - Login Successful!"

exit 0

else

echo "Invalid User. Try Again."

fi

attempt=$((attempt + 1))

done

echo "Maximum Attempts reached. Access Denied!"

**Out Put:**

Login Successful:

A black background with white text

AI-generated content may be incorrect.

Login Fail:

A screenshot of a computer

AI-generated content may be incorrect.

1. **Accept a string from terminal and echo suitable message if it does not have atleast 10 characters.**

echo "Enter a string: "

read str

length=${#str}

if [ $length -lt 10 ];

then

echo "The string is too short (Only $length characters). It must have at least 10 characters."

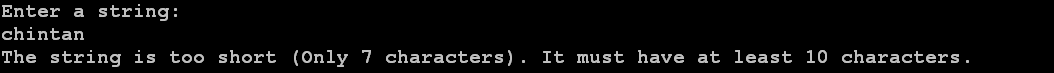
else

echo "Valid string! It has $length characters."

Fi

**Out Put:**

Invalid:



Valid:

A black background with white text

AI-generated content may be incorrect.

1. **Write a script to delete all vowels from particular string.**

echo "Enter a String:"

read str

result=$(echo "$str" | tr -d 'aeiouAEIOU')

echo "String without vowels: $result"

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Write a script to accept a number from user until he enters 0 & find sum of all that numbers.**

sum = 0

while true

do

echo -n "Enter a number (0 to Stop): "

read num

if [ $num -eq 0 ];

then

break

fi

sum=$((sum + num))

done

echo "Sum of all entered number = $sum"

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Write a script to check whether file is empty or not, and if file name doesn’t exist than print appropriate message.**

echo "Enter file name: "

read filename

if [ ! -e "$filename" ];

then

echo "File does not exist."

elif [ ! -s "$filename" ];

then

echo "File exists but it is empty."

else

echo "File exitsts and it is not empty."

Fi

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Write a shell script to prompts the user to enter the time (in 24-Hour format) then wish them “Good morning”, “Good Afternoon”, “Good Evening”, or “Good night” based on the input. (example: if Input is 13 then print Good afternoon)**

echo "Enter time in 24-horu format (0-23):"

read hour

if [ $hour -ge 0 ] && [ $hour -lt 12 ];

then

echo "Good Morning"

elif [ $hour -ge 12 ] && [ $hour -lt 17 ];

then

echo "Good Afternoon"

elif [ $hour -ge 17 ] && [ $hour -lt 21 ];

then

echo "Good Evening"

elif [ $hour -ge 21 ] && [ $hour -lt 24 ];

then

echo "Good Night"

else

echo "Invalid hour entered! Please enter a value between 0 to 23."

Fi

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Write a shell script which takes input of file name and prints first 10 lines of that file. File name is to be passed as command line argument. If argument is not passed then any ‘C’ program from the current directory is to be selected.**

if [ $# -eq 0 ];

then

file=$(ls \*.c 2>/dev/null | head -n 1)

if [ -z "$file" ];

then

echo "No C Program found in the current directory!"

exit 1

else

echo "No aregument provide. Using file: $file"

fi

else

file=$1

fi

if [ -f "$file" ];

then

echo "First 10 lines of $file: "

head -n 10 "$file"

else

echo "File '$file' does not exist!"

fi

**Out Put:**

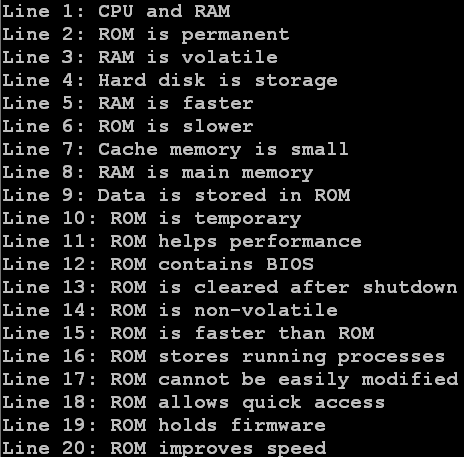
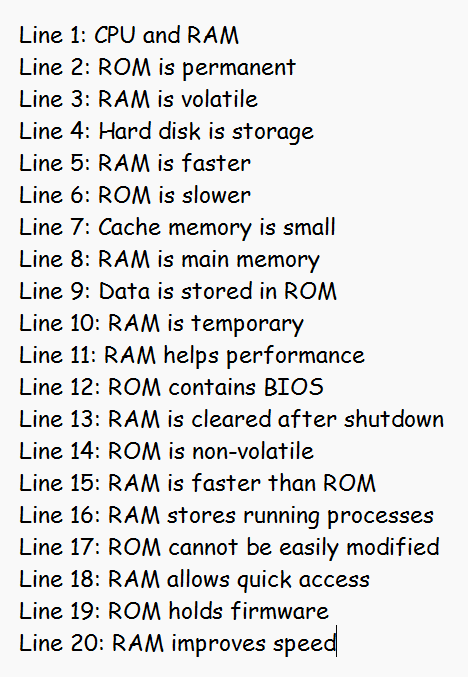
I can’t provide Out Put because I don’t have Possible File and Directory.

**Write Command Using sed Or grep**

1. **Write command to replace ‘RAM’ with ‘ROM’ on line no 10 to 20.**

sed '10,20 s/RAM/ROM/g' filename.txt

**Out Put:**



1. **Display all blank lined between line 20 and 30 of file test.txt.**

sed -n '20,30p' test.txt | grep '^$'

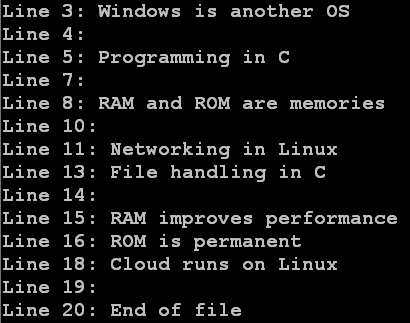
1. **To list file names consist of only 4 digits.**

ls | grep '^[0-9]\{4\}$'

1. **Display the lines that do not contain “Unix”.**

grep -v "Unix" test.txt

**Out Put:**



1. **Display the lines which are starting with 1 at the beginning.**

grep '^1' test.txt

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Display lines beginning either with alphabet or digit from file test.txt.**

grep '^[A-Za-z0-9]' test.txt

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **Write a command to display all file name containing only digits in a filename.**

ls | grep '^[0-9]\+$'

1. **Display two lines starting from 4th line of file test.txt.**

sed -n '4,5p' test.txt

**Out Put:**

**A black background with white text

AI-generated content may be incorrect.**

1. **To display lines beginning with numbers of a file y1.txt.**

grep '^[0-9]' y1.txt

**Out Put:**

A black background with white text

AI-generated content may be incorrect.

1. **To count number of words in line 10 thought 20 of file test.txt.**

sed -n '10,20p' test.txt | wc -w

**Out Put:**

**A black background with white text

AI-generated content may be incorrect.**

**Write The Command**

1. **Count number of characters in first five lines of file x1.**

head -n 5 x1 | wc -m

1. **Display files of current directory whose 1st character is not digit.**

ls | grep '^[^0-9]'

1. **Display last 2 lines of working directory.**

ls | tail -n 2

1. **Display only those files of current directory which is own by the current user.**

ls -l | grep "$USER"

1. **To combine content of two file do not use cat command.**

sed '' file1 file2

1. **Count the total no of blank lines of file x1.**

grep -c '^$' x1

1. **Display the lines which are not starting with 2 at the beginning.**

grep -v '^2' x1

1. **Count the total no. of lines in a file.**

wc -l x1

1. **To display lines beginning with alphabets of a file test.txt**

grep '^[A-Za-z]' test.txt

1. **Display lines of file line 3-5.**

sed -n '3,5p' x1