Vaishnavi Kuldharme

1. Write a shell script which will generate the O/P as follows

\*

\*\*

\*\*\*

\*\*\*\*

[admin@hostname01 ~]$ nano star

for((i=1; i<=4; i++))

do

for((j=1; j<=i; j++))

do

echo -n "\*"

done

echo

done

[admin@hostname01 ~]$ chmod +x star

[admin@hostname01 ~]$ ./star

\*

\*\*

\*\*\*

\*\*\*\*

[admin@hostname01 ~]$

2.Accept the first name, middle name, and last name of a person in variables fname, mname and lname respectively. Greet the person (take his full name) using appropriate message.

[admin@hostname01 ~]$ nano greet.sh

read -p "Enter The First Name: " First\_name

read -p "Enter The Middle Name: " Middle\_name

read -p "Enter The Last Name: " Last\_name

echo "Hello $First\_name $Middle\_name $Last\_name Have a good Day."

[admin@hostname01 Desktop]$ vim greet

[admin@hostname01 Desktop]$ chmod +x greet

[admin@hostname01 Desktop]$ ./greet

Enter The First Name: Vaishnavi

Enter The Middle Name: Uttam

Enter The Last Name: Kuldharme

Hello Vaishnavi Uttam Kuldharme name Have a good Day

3.Display the name of files in the current directory along with the names of files with maximum & minimum size. The file size is considered in bytes.

🡪 admin@hostname01 ~]$ nano file\_Size

#!/bin/bash

ls -lS --block-size=1 | awk '{print $5, $9}' | tail -n +2

max\_file=$(ls -S | head -n 1)

min\_file=$(ls -Sr | head -n 1)

max\_size=$(stat -c%s "$max\_file")

min\_size=$(stat -c%s "$min\_file")

echo "File with Max size: $max\_file ($max\_size bytes)"

echo " File with Min size: $min\_file ($min\_size bytes)"

[admin@hostname01 Desktop]$ vim file\_Size

[admin@hostname01 Desktop]$ chmod +x file\_Size

[admin@hostname01 Desktop]$ ./file\_Size

6067 .bash\_history

357 file\_Size

96 starttt

25 friends

20 demo

15 star

10 newfriend

6 start

File with Max size: .bash\_history (6067 bytes)

File with Min size: start (6 bytes)

1. Write a script which when executed checks out whether it is a working day or not?

(Note: Working day Mon-Fri)

admin@hostname01 ~]$ nano weather.sh

#!/bin/bash

# user input for the day

read -p "Enter day of week to check: " day

# Convert the input to lowercase for case-insensitive comparison

day=$(echo "$day" | tr '[:upper:]' '[:lower:]')

if echo "$day" | grep -iqE "^(mon|tue|wed|thu|fri)$"; then

echo "$day is working Day."

else

echo "$day is not working Day."

fi

[admin@hostname01 ~]$ chmod +x weather.sh

[admin@hostname01 ~]$ ./weather.sh

Enter day of week to check:mon

mon is working Day.

1. Write a script that accepts a member into HP health club, if the weight of the person is withing the range of 30-250 Kgs.

🡪 [admin@hostname01 ~]$ nano Hp.sh

!/bin/bash

read -p "Enter your weight in kg: " weight

if [ "$weight" -lt 30 ] || [ "$weight" -gt 250 ]; then

echo "Oops your weight is not in range (30-250 kg) "

else

echo "Welcome to HP Health Club you are registered successfully."

fi

[admin@hostname01 ~]$ chmod +x Hp.sh

[admin@hostname01 ~]$ ./Hp.sh

Enter your weight in kg: 56

Welcome to HP Health Club you are registered successfully.

1. Write a shell script that greets the user with an appropriate message depending on the system time.

🡪 [admin@hostname01 ~]$ nano greet.sh

#!/bin/bash

read -p "Enter the hour between 0-23 only): " hour

if [ "$hour" -lt 0 ] || [ "$hour" -ge 24 ]; then

echo "Invalid input Please enter the number between 0 and 23."

exit 1

fi

# Greet based otime of day

if [ "$hour" -ge 5 ] && [ "$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!"

else

echo "Good Night!"

fi

[admin@hostname01 ~]$ chmod +x greet.sh

[admin@hostname01 ~]$ ./greet.sh

Enter the hour (0-23): 22

Good Night!

1. A data file file has some student records including rollno, names and subject marks. The fields are separated by a “:”. Write a shell script that accepts roll number from the user, searches it in the file and if the roll number is present - allows the user to modify name and marks in 3 subjects.   
   If the roll number is not present, display a message “Roll No Not Found”. Allow the user to modify one record at a time.

[admin@hostname01 ~]$ nano student.sh

#!/bin/bash

file="student\_records"

read -p "Enter the roll number to search: " rollno

record=$(grep "^$rollno:" "$file")

if [ -n "$record" ]; then

echo "Record found: $record"

IFS=":" read -r roll name marks1 marks2 marks3 <<< "$record"

# Allow user to modify name and marks for modification

read -p "Enter new name (current: $name): " new\_name

read -p "Enter new mark for subject 1 (current: $marks1): " new\_marks1

read -p "Enter new mark for subject 2 (current: $marks2): " new\_marks2

read -p "Enter new mark for subject 3 (current: $marks3): " new\_marks3

sed -i "s/^$rollno:$name:$marks1:$marks2:$marks3$/$rollno:$new\_name:$new\_marks1:$new\_marks2:$new\_marks3/" "$file"

echo "Record updated successfully!"

else

# If roll number is not found

echo "Roll No Not Found"

fi

[admin@hostname01 ~]$ ./student.sh

Enter the roll number to search: 2

Record found2:vaishnavi:60:75:97

Enter new name (current:vaishnavi): Om

Enter new mark for subject 1 (current: 60): 90

Enter new mark for subject 2 (current: 75): 95

Enter new mark for subject 3 (current: 97): 97

Record updated successfully

[admin@hostname01 ~]$ cat student\_records

1:Shivam:80:90:70

2:vaishnavi:60:75:97

3:Rohan:80:30:80

1. Modify program 7 to accept the RollNo from the command line.

🡪 [admin@hostname01 ~]$ nano student.sh

# To Accept roll number

If[-z "$1 "];then

echo "us:$0 <rollno>"

exit 1

fi

rollno=$1

1. Modify the program 7 to accept the RollNo and display the record and ask for delete confirmation. Once confirmed delete the record and update the data file.

[admin@hostname01 ~]$ nano del\_stu.sh

#!/bin/bash

file=" student\_records"

if [ -z "$1" ]; then

read -p "Enter the roll number to search: " rollno

else

# Use the command line argument for roll number

rollno=$1

fi

record=$(grep "^$rollno:" "$file")

if [ -n "$record" ]; then

echo "Record found: $record"

read -p "want to delete record type y : " confirm

if [ "$confirm" == "y" ] || [ "$confirm" == "Y" ]; then

sed -i "/^$rollno:/d" "$file"

echo "Record deleted successfully!"

else

echo "Deletion aborted."

fi

else

echo "Roll No Not Found"

fi

[admin@hostname01 ~]$ chmod +x del\_stu.sh

[admin@hostname01 ~]$ ./del\_stu.sh

Enter the roll number to search: 3

Record found3:Rohan:80:30:80

want to delete record type y : y

Record deleted successfully!

1. Write a script that takes a command line argument and reports on its file type (regular file, directory file, etc.). For more than one argument generate error message.

🡪 [admin@hostname01 ~]$ nano fileReport.sh

if [ "$#" -ne 1 ]; then

echo "Error provide exactly one argument."

echo "Usage: $0 <file\_path>"

exit 1

fi

file\_path=$1

if [ -e "$file\_path" ]; then

file\_type=$(file "$file\_path")

echo "$file\_type"

else

echo "Error: $file\_path does not exist."

exit 1

fi

1. Add some student records in the “student” file manually. The fields to be considered are “RollNo”, “Name”, “Marks\_Hindi”, “Marks\_Maths”, “Marks\_Physics”.

🡪 [admin@hostname01 ~]$ touch student

[admin@hostname01 ~]$ vim student

[admin@hostname01 ~]$ cat student

1:Shivam:80:90:70

2:vaishnavi:60:75:97

3:Rohan:80:30:80

[admin@hostname01 ~]$ nano studentRecord.sh

[admin@hostname01 ~]$ chmod +x studentRecord.sh

[admin@hostname01 ~]$ ./studentRecord.sh

Enter Roll No: 4

Enter Name: Omkar

Enter Marks in Hindi: 74

Enter Marks in Maths: 94

Enter Marks in Physics: 89

Total Marks: 257

Percentage: 85%

Write a script which does the following:

* 1. If the roll number already exists, then store the record and the following message   
     “roll number exists” in a log file “log1”.

🡪 [admin@hostname01 ~]$ ./studentRecord.sh

Enter Roll No: 2

Enter Name: vaishnavi

Roll number 2 exists

* 1. If the marks in the subjects is not in the range of 1 – 99 then store such a record followed by a message “marks out of range” in “log1”

🡪 [admin@hostname01 ~]$ ./studentRecord.sh

Enter Roll No: 6

Enter Name: Priya

Enter Marks in Hindi: 94

Enter Marks in Maths: 74

Enter Marks in Physics: 101

Marks out of range.

* 1. If the data is valid, the calculate total, percentage, grade and display on the terminal

🡪

[admin@hostname01 ~]$ ./studentRecord.sh

Enter Roll No: 8

Enter Name: siddhi

Enter Marks in Hindi: 96

Enter Marks in Maths: 34

Enter Marks in Physics: 61

Total Marks: 191

Percentage: 63%

# shell code for this :

#!/bin/bash

student\_file="student"

log\_file="log1"

read -p "Enter Roll No: " roll\_no

read -p "Enter Name: " name

read -p "Enter Marks in Hindi: " marks\_hindi

read -p "Enter Marks in Maths: " marks\_maths

read -p "Enter Marks in Physics: " marks\_physics

# for roll number already exists or not

if grep -q "^$roll\_no:" "$student\_file"; then

echo "Roll number $roll\_no exists."

echo "roll number exists" >> "$log\_file"

exit 1

fi

#checked for its in range or not

if [ "$marks\_hindi" -lt 1 ] || [ "$marks\_hindi" -gt 99 ] || [ "$marks\_maths" -lt 1 ] || [ "$marks\_maths" -gt 99 ] || [ "$marks\_physics" -lt 1 ] || [ "$marks\_physics" -gt 99 ]; then

echo "Marks out of range."

echo "marks out of range" >> "$log\_file"

exit 1

fi

echo "$roll\_no:$name:$marks\_hindi:$marks\_maths:$marks\_physics" >> "$student\_file"

total=$((marks\_hindi + marks\_maths + marks\_physics))

percentage=$((total / 3))

echo "Total Marks: $total"

echo "Percentage: $percentage%"