SAMPLE TEST

Part 1 – MCQ

1. What is the default shell in Linux?
   1. bash
   2. csh
   3. ksh
   4. zsh
2. Which line of command will resume the most recent process in the background?
   1. kill
   2. jobs
   3. fg
   4. bg
3. Which symbol is used to start the process in the background?
   1. ^
   2. $
   3. &
   4. \*
4. What does the following command do?

egrep ‘A+’ filename

* 1. Matches zero or more occurrences of ‘A’
  2. Matches one or more occurrences of ‘A’
  3. Matches zero or one occurrence of ‘A’
  4. Matches A followed by any character

1. Consider the file that contains the following :

Text

Description automatically generated

Which line(s) will be displayed when the following command is issued?

egrep ‘th.\*is’ test.file

* 1. Line 1 only
  2. Line 1 and 4
  3. Line 4 and 5
  4. Line 1 and 5
  5. None of the above

1. What will the following command print? sed -n ‘$!p’ test.file
   1. All lines printed twice
   2. Only the last line
   3. All except the last line
   4. None of the above
2. What is the regular expression that selects all lines that begin with 2 or more occurrences of the word “this”?
   1. “^(this) (this )+”
   2. “(this this)?”
   3. “(this )+”
   4. “^(this ){2,}”
   5. Both a and d
3. What is the regular expression that selects all empty lines?
   1. ‘\*’
   2. ‘\b’
   3. ‘$^’
   4. ‘^$’ //頂尾
4. Which command will display detailed information about **only** the hidden files and directories in the current directory?
   1. ls -ld .\*
   2. ls -al
   3. ls -la | grep “.”
   4. ls -lah
5. what will the following command print? sed ‘s/0/!/2g’ cars.txt
   1. replace all occurrences of 0 with !
   2. replace only the first two occurrences of 0 with !
   3. starting at the second instance of 0, replace all 0 with !
   4. replace only the second instance of 0 with !

Part 2 – short answers

1. what is the regular expression that selects all lines that begin with 2 or more occurrences of an upper case alphabet, immediately followed by 3 whitespaces, and one or two occurrences of the word this?
   1. “^[A-Z]{2,}\s{3}(this ){1,2}”
2. What is the regular expression that selects all lines that contain 2 to 5 occurrences of “a”, followed by any number of any characters, 0 or 1 occurrence of “x”, any number of any characters and ends with either a digit or the word “bye”?
   1. “a{2,5}.\*x?.\*([0-9]|bye)$”
3. Using sed utility, display lines from 1 to 5, and 9 to 20, replacing all ‘|’ with two colons :: from a file called cars.txt.
   1. sed -n 's/|/::/g; 1,5 p; 9,20 p' cars.txt
   2. sed 's/|/::/g; 20q; 6,8d' cars.txt
4. Using sed utility, display only the cars that are less than $70000 from cars.txt. HINT: price is the last column.
   1. sed -n '/[0-6][0-9][0-9][0-9][0-9]$/ p' cars.txt
   2. sed '/[7-9][0-9][0-9][0-9][0-9]$/ d' cars.txt
5. Using awk utility, display the following from employees.txt : “Employee <full name> has the total compensation of $<amount>” where amount is the annual salary (4th column) plus the bonus (5th column). HINT: first name is in the second column, and last name is in the third column.
   1. awk '{total=$4+$5} {print "Employee", $2, $3, "has the total compensation of $" total}' employees.txt
6. Using awk utility, from cars.txt file, if the price of the car is more than 35000, display the following:
   1. awk -F"|" '$5 > 35000 {print "Model: " $2 "\nMake: " $1 "\nYear: " $3 "\nPrice: " $5}' cars.txt

Model: <model name>

Make: <make>

Year: <year>

Price: <price>

1. Using awk utility and employees.txt file, provide a command that produces the following output :
   1. awk 'BEGIN {print "\*\*\*\*PROCESSING\*\*\*\*"} {salary+=$4;bonus+=$5} END {print "TOTAL SALARY: " salary "\nTOTAL BONUS: " bonus "\n\*\*\*\*COMPLETED\*\*\*\*"}' employees.txt

Text

Description automatically generated

Where the total salary is the sum of all values in the fourth column

1. Using awk utility, find the employee with the highest salary (salary + bonus) from employees.txt and display the following:
   1. awk 'max<$4 {max=$4; fname=$2;lname=$3;total=$4+$5} END {print "Employee " fname, lname, "makes the highest salary of $" total}' employees.txt

Employee <full name> makes the highest salary of $<salary+bonus>

Part 3 – bash scripting

Create a bash script called grades.bash that does the following:

1. Prompts user 5 times for grades in each course.
2. Compute the sum of 5 grades, and the average.
3. Display the total grade and average.

SAMPLE OUTPUT

Text

Description automatically generated

#!/bin/bash

sum=0

for i in {1..5}

do

read -p "Enter grade for subject #$i: " num

sum=$(($sum + $num))

done

let average=$sum/$i

echo Your total grade is $sum

echo Your average is $average

Create a bash script called numbers.bash that does the following:

1. Clears the screen.
2. Prompts user for the input (a number less than 10)
3. Prints error message for incorrect input (containing non-digit characters)
4. Also prints error message for number greater than or equal to 10.
5. Continue to prompt until valid input is entered.
6. If the input is valid, then print the output as shown below:

SAMPLE OUTPUT

Text

Description automatically generated

#!/bin/bash

valid=false

clear

while [[ "$valid" = false ]]

do

read -p "Type in a number less than 10: " num

if [ $(echo $num | grep "[^0-9]") ]

then

echo "Incorrect data input!"

elif [[ num -lt 1 || num -gt 9 ]]

then

echo "Number must be between 1 and 9!"

else

echo "You entered $num"

valid=true

fi

done

for ((i=$num;i>0;i--))

do

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

do

echo -n "$i"

done

echo

done

for ((i=2;i<=$num;i++))

do

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

do

echo -n "$i"

done

echo

done

Create a bash script called palindrome.bash that does the following

1. Prompt the user for a word, or a sentence
2. Make sure all white spaces are removed.
3. Check to see if the word is palindrome after trimming off whitespaces.
   1. palindrome is a word that is identical when reversed.
   2. e.g. race car, radar, rotator, civic, bob, eve, dad, was it a car or a cat I saw

SAMPLE OUTPUT

Text

Description automatically generated

#!/bin/bash

read -p "Enter a word: " word

nw=$(echo $word | tr [A-Z] [a-z] | tr -d ' ')

//tr" command removes any whitespace in the word using the "-d" option.

bw=$(echo $nw | rev)

// reversed using the "rev" command and stored in the "bw" variable.

if [ $nw == $bw ]

// strings comparison use ==

then

echo $word is a Palindrome

else

echo $word is not a Palindrome

fi