**ULI101 2022 – MOCK TEST – SLG SESSIONS**

**MCQs (10 X 2 = 20)**

1. Without the -n option, what will the following sed command print to the terminal? | sed '/line/' readme.txt
2. All lines that include 'line'
3. All lines and the lines including "line" (some duplicates)
4. Will print the line numbers
5. All the lines in the file
6. What will the following command print? | sed -n '4,20 p' readme.txt
7. All lines and lines 4-20 (duplicates)
8. Only lines 4-20
9. Only line 4 and 20
10. Lines 5-19
11. What will the following command do? | sed 's/hello/goodbye/' readme
12. Find the first instance of hello in each line and replace it with goodbye
13. Add hello after every goodbye in the file
14. Search for "hello" and "goodbye" and print to screen
15. Replace all instances of "hello" with "goodbye"
16. What will the following command do? | sed 's/hello/goodbye/g' readme
17. Substitute all instances of "hello" with "goodbye" and print to screen
18. Find the first instance of "hello" in each line and replace it with "goodbye" and print to screen
19. Substitute all instances of "hello" and replace it with "goodbye" but write the changes to the file
20. Shut down the computer and print Hello Goodbye
21. what would be the result of the following command: awk '$3 ~/5$/ {print $3, $1, "$" $5}' c*ars*
22. printing column 3 [space] column 1 [space] column 5
23. printing column 3 [space] column 1 [space] $column 5
24. printing column 3 column 1 $column 5
25. printing column 3 column 1 column 5
26. What does “X?” mean in the regular expression?
27. Matches zero or more capital X’s
28. Matches no or one occurrence of the capital letter X.
29. Matches one or more capital X’s
30. All of the Mentioned
31. What does the following segment of code does?:

grep -i man heroes.txt

1. Manually opens a file called heroes.txt
2. Manages heroes.txt
3. Search for “man” in the file “heroes.txt”
4. None of the mentioned
5. Which of the following options support the given statement?

Statement: A regular expression could be a fixed word or describe something like more general

1. This flexibility makes Regular expression valuable
2. This flexibility makes the Regular expression unvaluable
3. Both (a) and (b)
4. None of the mentioned
5. State true or false:

Statement: A Regular expression is a sequence of characters that represent a pattern

1. True
2. False
3. Generate the regular expression to match blank lines
4. /\*/
5. /bl
6. /^?/
7. /^$/

**# SHORT ANSWER QUESTIONS (5 X 5 = 25)**

- You have the following "pi\_data.txt" which has the following content in it:

Text

Description automatically generated

You can get this file using the following command

cp ~ hsehjal/slgFall\_2022/mockTest/pi\_data.txt ./pi\_data.txt

1. Write a Linux awk command to display only the lines whose wave(ft) is higher than 4 for file “pi\_data.txt”.

(\*IMP: Add a write representation as shown below just awk command only. DO NOT use echo or any command for this)

**EXPECTED OUTPUT:**

**Text

Description automatically generated**

2. Write a Linux pipeline command to display only time, wave(ft) value, and then "small"(if wave value is less than 4) or "medium"(if wave value is greater than or equal to 4) for file “pi\_data.txt”

**EXPECTED OUTPUT:**

**Text

Description automatically generated**

3. Explain how this script works and what it does. Give 2 or 3 examples of invoking this script.

./letArguments.bash

Text

Description automatically generated

4. Consider the following 2x scripts :

./hamit.bash :

A picture containing text

Description automatically generated

./tarik.bash :

Text

Description automatically generated

* What is the role of the export statement ? and what would be the output? Explain.
* What would be the output if we remove the “export var” from “hamit.bash” file and then run it

5. You are provided with this script "doordash.bash" !!

Text

Description automatically generated

**SHOW and EXPLAIN what the result would be displayed upon entering each of the following, in the order shown:**

A. $ ./doordash.bash Sushi Tacos Burrito

B. $ echo $?

C. $ ./doordash.bash Tacos Burrito

D. $ ./doordash.bash

**# Let's WRITE some SCRIPTS (10 X 3 = 30):**

**1. Write a bash script named "showMyGrade.bash" that automates the following tasks for us:**

a. clears the screen

b. Prompt student to enter their mark. You can use this template "Please enter your percentage Mark (0-100):"

c. Store the value of user-input in a variable. It's a good practice to give meaningful names to the variable

d. Based on the following criteria, display the User's grade in a nice representation!!

- mark in the range 90-100, grade is A

- mark in the range 81-89, grade is B

- mark in the range 71-80, grade is C

- mark in the range 65-70, grade is D

- mark in the range 55-64, grade is E

- mark less than 55, grade is FAIL

(\*IMP: range is inclusive for both lower and upper limit)

**2. Let's Update the Script "showMyGrade.bash" for User Validation. We will validate the User Input to only allow input that is a \*DIGIT\* lying in the range 0-100. That means our script will keep asking for input until a valid \*DIGIT\* in the range (1-100) is entered. Update your script to give a nice error whenever user inputs a wrong input.**

(\*IMP: Your script should keep asking for input until user enters a valid input)

**3. BONUS QUESTION: Write a bash script named "play.bash" that clears the screen and then display the following pattern on the screen:** (HINT: for-loop)

Text

Description automatically generated