Command Line Assignment

Peer Learning Document

Questions:

Question 1. Write a bash script to get the current date, time, username, home directory and current working directory.

Question 2. Write a bash script (name Table.sh) to print the Table of a number by using a while loop. It should support the following requirements.

- The script should accept the input from the command line.

- If you don't input any data, then display an error message to execute the script correctly.

Question 3. Write a Function in bash script to check if the number is prime or not? It should support the following requirement.

- The script should accept the input from the User.

Question 4. Create a bash script that supports the following requirement.

- Create a folder 'Assignment'.

- Create a file 'File1.txt' inside 'Assignment' Folder.

- Copy all the content of Table.sh(2nd script) in 'File1.txt' without using 'cp' and 'mv' command.

- Append the text Welcome to Sigmoid' to the 'File1.txt' file.

- List all the directories and files present inside Desktop Folder.

Question 5. You have given an array. Using Bash script, print its length, maximum element and minimum element of arr=(2 3 4 1 6 7).

Karan’s Solutions

Question-1:

He has used the following commands for displaying the required parameters asked in the question.

$(date)	Command to fetch the date
$(date +%T)	Command to fetch the time
$(whoami)	Command to fetch the current working user
$HOME	Command to fetch the Home directory
$(pwd)	Command to fetch the current working directory

These commands are the same as mine.

Question-2:

His script takes user input for a number and checks if it is a valid input or not using a case statement. If the input is a number, it generates the multiplication table for that number from 1 to 10. If the input is not a number, it displays an error message. If there is no input, it also displays an error message.

How is his solution different from mine?

- Karan displaying the user to enter the input, whereas my solution expects the arguments passed while running the script itself.
- In my solution user can pass multiple arguments and get multiple tables as a result.
- He is also validating user input whether the passed value is number or not.

Question-3:

His code takes user input for a number and checks if it is a valid input or not using a case statement. If the input is a valid number, it calls a function "check_prime" to determine if the number is prime or not. If the input is not a valid number, it displays an error message. If there is no input, it also displays an error message.
The "check_prime" function takes a number as input and checks if it is prime or not by using a for loop to divide the number by all the integers from 2 to half the value of the

number. If the number is divisible by any of these integers, it is not a prime number. Otherwise, it is a prime number.

His solution is logically the same as mine, just some minor implementations are different.

Question-4:

His script creates a folder called "Assignment" in the current directory using the "mkdir" command. It then creates a file called "File1.txt" inside the "Assignment" folder using the "touch" command.
The content of the "table.sh" file is then copied to "File1.txt" using the "cat" command.
The text "Welcome to Sigmoid" is then appended to the end of "File1.txt" using the "echo" command.
Then He is listing all the directories and files present inside the Desktop folder and appends the output to a text file called "DesktopListDirectories.txt" using the "ls" command with the "-al" option.
Finally, he is opening the "DesktopListDirectories.txt" file using the "open" command. This command may not work on all systems, as it depends on the operating system and the default file viewer application.

How is his solution different from mine?

- I created the assignment folder in the desktop whereas karan created it in his current working directory.
- Other things are the same as mine.

Question-5:

His code prompts the user to enter the size of an array and reads the input using the "read" command. It then creates an empty array and prompts the user to enter each element of the array using a loop. The input is read using the "read" command and added to the array using array indexing.
Then it prints the total number of elements in the array using the "${#arr[@]}" notation, and the array elements using "${arr[@]}".
He defined a function called "max_min_ele" to find the maximum and minimum elements in the array. The function initializes the "max" and "min" variables to the first element of the array, and then loops over the remaining elements, updating the "max" and "min" variables as necessary. The function then prints the maximum and minimum elements using the "echo" command.

Finally,He is calling the "max_min_ele" function to find the maximum and minimum elements in the array.

How is his solution different from mine?

- I am finding the length, max, min from the array given in the question, whereas karan is taking the array as user input.
- He is finding the max and min by creating a function separately.

Akshay's Solution

Question-1:

The commands used are the same as mine and karan's.

Question-2:

- His script prompts the user to enter a number and reads the input using the read command.
- Then he checks if the input is empty or not using the -z operator. If the input is empty, it displays a message "No input entered" and exits.
- Then he is checking if the input is a number or not using a regular expression. If the input is not a number, it displays an error message "Not a number, invalid input" and exit.
- If the input is a number, the he is calculating multiplication table using a 'while' loop

How is his solution different from mine?

- Akshay is displaying the user to enter the input, whereas my solution expects the arguments passed while running the script itself.
- In my solution users can pass multiple arguments and get multiple tables as a result.

Question-3:

His script takes user input for a number and checks whether it's a prime number or not. Here's a brief overview of how the script works:

1. It first prompts the user to enter a number using echo and then reads the user input using the read command.
2. It then checks whether the user has entered any input or not using if [[-z "$n"]]; then and prints a message "No input entered" if the user has not entered any input.
3. It checks whether the user input is a number or not using regular expressions.

4. After all validations, it checks if the number is prime or not.

His solution is logically the same as mine. But the only difference is I am using a function for checking the prime number.

Question-4:

The commands are almost identical to mine and karan's.

Question-5:

His program starts by prompting the user to enter an array, and then reads in the user input using the read command with the -a option, which reads the input as an array.
The program then initializes the max and min variables to the first element of the array. These variables will be used to keep track of the maximum and minimum values in the array.
He then loops through all the elements in the array using a for loop that iterates over the elements in the arr variable. For each element, the program checks whether it is greater than the current value of max, and if so, updates the max variable. Similarly, the program checks whether the element is less than the current value of min, and if so, updates the min variable.
Thus he is finding the max and min values of the array.

How is his solution different from mine?

- I am finding the length, max, min from the array given in the question, whereas Akshay is taking the array as user input.