System Programming

M-Tech refresher module

Lab Assignmnet-2 **Deadline:** 25.07.23

Write the **assembly code of question no 1 and 2.**

1. Create a program to iteratively find the nth Fibonacci number. The value for n should be set as a parameter (e.g., a programmer defined constant).

The formula for computing Fibonacci is as follows: fibonacci(n) = { n if n=0 or n=1 fibonacci(n−2) + fibonacci(n−1) if n≥2

i.Correct code: (marks:10)

ii. Use the debugger to execute the program and display the final results. (marks: 5)

iii. Test the program for various values of n (mark:5)

2. Simple example program to convert an integer into an ASCII string

Correct code: (marks:10)

Use the debugger to execute the program and display the final results. (marks: 5)

Test the program for various values of n (mark:5)

3. Write a c program tail -n which will print last n lines of the input. The program should behave rationally no matter how much the value of n should be. Do not store the lines in 2-dimentional arrays of fixed sizes.

For correct code and execution (marks: 15)

4. Write a script that will display the chessboard on the screen (marks10)

5. **File Sorting** (marks: 15)

Instructions:

Write a shell script or command-line program to perform the following tasks.
Use appropriate command-line arguments or prompts to receive inputs and display outputs.
Document your code with comments to explain the purpose and functionality of each section.
Tasks:

Prompt the user to enter the name of a directory.
Check if the directory exists. If it doesn't, display an error message and exit the program.
List all the files in the given directory.
Sort the files alphabetically.

Create a new directory named "sorted" inside the given directory.
Move each file from the original directory to the "sorted" directory.
Display a success message with the total number of files moved.
Note: Ensure proper error handling and informative error messages throughout the code.

Submission:

Write the code in a file named "file_sorting.sh" (for shell script) or "file_sorting.py" (for Python script).
Include any necessary instructions or explanations as comments within the code.
Provide a brief explanation of your approach

6. You are given a directory named "logs" that contains a set of log files. Each log file has a name in the format “log_YYYYMMDD.txt”, where "YYYY" represents the year, "MM" represents the month, and "DD" represents the day. The log files contain entries in the following format:

Directory: log_folder
Download this folder, unzip it, and then perform the following tasks.
Write a Linux command or script that performs the following tasks:
1. Reads all log files in the "logs" directory.
2. Extract the timestamp and message from each log entry.
3. Filter out log entries that have a timestamp older than a given date.
4. Sort the remaining log entries in descending order based on their timestamps.
5. Writes the sorted log entries to a new file named "filtered_logs.txt" in the following format:
Timestamp: <timestamp_value>
Message: <log_message>
Timestamp: <timestamp_value>
Message: <log_message>
…
6. Calculates the average time difference between consecutive log entries for each log file.
7. Find the log file with the maximum average time difference.
8. Print and save the output filename and corresponding maximum average time difference in the following format:
Expected output:
Filename: log_20220102.txt
Maximum Average Time Difference: 300 seconds.
9. Find the log file with the longest average message length.
10. Print and save the output filename and corresponding longest average message length in the following format:
Filename: log_20220101.txt
Longest Average Message Length: 21 characters

Submission: folder containing all three text files, a word file that explains every step, screenshots of every question, Linux commands, and a readme file. (marks:15)

Note: Submit all your code in word files over the google classroom before deadline.