Linux Assignment

Name: Debojyoti Mishra
Class: L1, Computer Science, 2023-2026

Problem 1:

debojyotimishra — nano user_audit.sh — 80×57

```
Last login: Thu Dec 14 11:40:15 on ttys001
debojyotimishra@Debojyotis-MacBook-Pro ~ % nano user_audit.sh
```

```
  UW PICO 5.09                        File: user_audit.sh

output_file="user_list.txt"

if [ "$#" -eq 1 ]; then
    output_file="$1"
fi

usernames=$(extract_usernames)

save_sorted_usernames <<<"$usernames" "$output_file"

echo "User list saved to $output_file"
```

Problem 2:

debojyotimishra — nano count_errors.sh — 80×63

```
debojyotimishra@Debojyotis-MacBook-Pro ~ % nano count_errors.sh
debojyotimishra@Debojyotis-MacBook-Pro ~ % chmod +x count_errors.sh
debojyotimishra@Debojyotis-MacBook-Pro ~ % ./count_errors.sh "ERROR: Something w
ent wrong" "/Users/debojyotimishra/count_errors.sh"
debojyotimishra@Debojyotis-MacBook-Pro ~ % nano count_errors.sh
```

```
  UW PICO 5.09                    File: count_errors.sh

#!/bin/bash

# Function to count occurrences of error message in log file
count_error_occurrences() {
    error_message="$1"
    log_file="$2"

    # Use grep to find lines containing the error message and count the occurre$
    count=$(grep -c "$error_message" "$log_file")

    echo "The error message '$error_message' occurred $count times in $log_file"
}

# Check if correct number of arguments are provided
if [ "$#" -ne 2 ]; then
    echo "Usage: $0 <error_message> <log_file>"
    exit 1
fi
```

Problem 3:

debojyotimishra — nano monitor_process_cpu.sh — 80×61

```
  UW PICO 5.09                    File: monitor_process_cpu.sh

#!/bin/bash

# Function to monitor process CPU usage and take action if it exceeds the thres$
monitor_process_cpu() {
    process_name="$1"
    cpu_threshold="$2"

    # Get the PID of the process using its name
    pid=$(pgrep "$process_name")

    if [ -z "$pid" ]; then
        echo "The process '$process_name' is not running."
        exit 1
    fi

    # Get CPU usage of the process
    cpu_usage=$(ps -p "$pid" -o %cpu=)

    echo "Current CPU usage of process '$process_name' (PID: $pid): $cpu_usage%"

    # Check if CPU usage exceeds the threshold
    if (( $(echo "$cpu_usage > $cpu_threshold" | bc -l) )); then
        echo "CPU usage exceeds the threshold of $cpu_threshold%"

        # Stop the process gracefully
        kill "$pid"

        echo "Process '$process_name' (PID: $pid) has been stopped."
    else
        echo "CPU usage is within the threshold."
    fi
}

# Check if correct number of arguments are provided
if [ "$#" -ne 2 ]; then
    echo "Usage: $0 <process_name> <cpu_threshold>"
    exit 1
fi

# Call the function with command line arguments
monitor_process_cpu "$1" "$2"

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

debojyotimishra — nano monitor_process_cpu.sh — 80×61

```
Last login: Wed Jan  3 09:46:41 on ttys001
[debojyotimishra@Debojyotis-MacBook-Pro ~ % nano monitor_process_cpu.sh          ]

debojyotimishra@Debojyotis-MacBook-Pro ~ % chmod +x monitor_process_cpu.sh

debojyotimishra@Debojyotis-MacBook-Pro ~ % ./monitor_process_cpu.sh "Google Chro
me" 50

ps: Invalid process id: 1410
5864
6795
6796
7223
7243
7267
7344
52655
81428
87422
96900
96907
96908
96909
96914
96917
96919
96929
96932
96933
96934
96935
96952
Current CPU usage of process 'Google Chrome' (PID: 1410
5864
6795
6796
7223
7243
7267
7344
52655
81428
87422
96900
96907
96908
96909
96914
96917
96919
96929
96932
96933
96934
96935
96952): %

Parse error: bad token
    <stdin>:1

CPU usage is within the threshold.
```

Problem 4:

debojyotimishra — nano generate_disk_usage_report.sh — 81×10

```
debojyotimishra@Debojyotis-MacBook-Pro ~ % nano generate_disk_usage_report.sh

debojyotimishra@Debojyotis-MacBook-Pro ~ % chmod +x generate_disk_usage_report.sh

debojyotimishra@Debojyotis-MacBook-Pro ~ % nano generate_disk_usage_report.sh
```

debojyotimishra — nano generate_disk_usage_report.sh — 81×48

```
  UW PICO 5.09                File: generate_disk_usage_report.sh

#!/bin/bash

# Function to generate disk usage report and append to specified output file
generate_disk_usage_report() {
    output_file="$1"

    # Use df command to get disk usage information and sort the results by disk $
    df -h | sort -k5 -h > "$output_file"

    echo "Disk usage report has been generated and saved to '$output_file'"
}

# Check if an output filename is provided, if not, set default filename
if [ "$#" -eq 0 ]; then
    output_file="disk_usage_report.txt"
else
    output_file="$1"
fi

# Call the function to generate the disk usage report
generate_disk_usage_report "$output_file"

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Pg   ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where is  ^V Next Pg   ^U UnCut Text^T To Spell
```

Problem 5:

debojyotimishra — nano log_rotation.sh — 80×47

```
  UW PICO 5.09                  File: log_rotation.sh                    Modified

#!/bin/bash

# Function to perform log rotation based on size threshold
perform_log_rotation() {
    log_file="$1"
    size_threshold="$2"

    # Check if the log file exists
    if [ ! -f "$log_file" ]; then
        echo "Error: Log file '$log_file' does not exist."
        exit 1
    fi

    # Get the current size of the log file
    current_size=$(du -b "$log_file" | awk '{print $1}')

    # Check if the log file size exceeds the threshold
    if [ "$current_size" -gt "$size_threshold" ]; then
        # Rename the existing log file with a version number
        mv "$log_file" "${log_file}.1"

        # Create a new empty log file with the original name
        touch "$log_file"

        echo "Log file '$log_file' exceeded the size threshold."
        echo "Rotated old log file to '${log_file}.1' and created a new log fil$
    else
        echo "Log file '$log_file' size is within the threshold."
    fi
}

# Check if correct number of arguments are provided
if [ "$#" -ne 2 ]; then
    echo "Usage: $0 <log_file> <size_threshold>"
    exit 1
fi

# Call the function with command line arguments
perform_log_rotation "$1" "$2"

^G Get
^X Exit
```

debojyotimishra — -zsh — 80×6

```
debojyotimishra@Debojyotis-MacBook-Pro ~ %
debojyotimishra@Debojyotis-MacBook-Pro ~ % nano log_rotation.sh

debojyotimishra@Debojyotis-MacBook-Pro ~ % chmod +x log_rotation.sh

debojyotimishra@Debojyotis-MacBook-Pro ~ %
```