Assignment -2 SHELL CODE

Akhil P Dominic
MT23013

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

chess-1.sh

```
#!/bin/sh

for(( i=0;i<8;i++))                             #outer for loop
do
        for((j=0;j<8;j++))                      #inner for loop
        do
                val=$(((i+j)%2))                #calculating the sum of i and j
                if [ $val -eq 0 ]           #checking if the value val is even
                then
                        echo -e -n "\u2588"     #If even , printing white
                else
                        echo -e -n " "          #else, printing black
                fi
        done
        echo " "
done
```

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.

file_sorting.sh

```
#!/bin/bash
echo "enter the full path of the directory name"                    #prompting user to enter full path
read directory_name

#reading directory name

if [ -d "$directory_name" ];
#checking if directory name exists

then
        sorted_f=$(ls $directory_name | sort)                        #sorting the files in the directory
        mkdir -p sorted
        #making a directory named sorted
```

```
        count=0
        echo "The files in the directory are : "
        echo "$(ls -l $directory_name)"                                                    #listing
all files in the directory
        for i in $sorted_f;
#looping through all files in the directory
        do
                mv $directory_name/$i sorted
#moving all files to sorted directory
                count=$(( count+1 ))
#increment count
        done

        if [ $count -eq 0 ]
#checking if directory is empty
        then
                echo "No files in the directory to move"
        else
                echo "Success !!!"
        #printing success message
        fi
        echo "Moved $count files"

else
        echo "$directory not found"
#printing directory not found message
fi
```

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:

log_prog.sh

```
log_dir="./logs"                                        #initialising the log
directory

logs=$(ls "$log_dir")                                   #listing all files in log_dir

date_num=$(echo "$(date +%F)")                          #saving current date
cur_date=$(echo "${date_num:8:10}")                     #Extracting the date day
value from the whole date


for log in $logs;                                       #looping through all
dates
do
        count=0
        log_array=()
        for word in $(cat "$log_dir/$log");             #Looking at each word form the log
        do

                if [ $count == 1 ]
```

```
                then
                        echo ${word:8:10}>bing_text
                        consider_word=${word:8:10}                      #Taking only the date day
value from the log

                        if [ $cur_date -gt $consider_word ]
                        then
                                echo "Filtering file.."                 #Filtering out all the dates
which falls below the current date day
                                rm "$log_dir/$log"
                        fi
                fi

                if [ $count -eq 3 ]
                then
                        log_array+=$(echo -e " \n ")
                fi
                log_array+=$(echo "$word ")
                count=$((count+1))


        done
        echo "$log_array   "
        echo -e " \n "


done
        cur_date=$(date)
        #date_val=$(date -f <(echo "$cur_date") "+%Y-%m-%d %H:%M:%S")
        #echo "Current date : $cur_date"
```