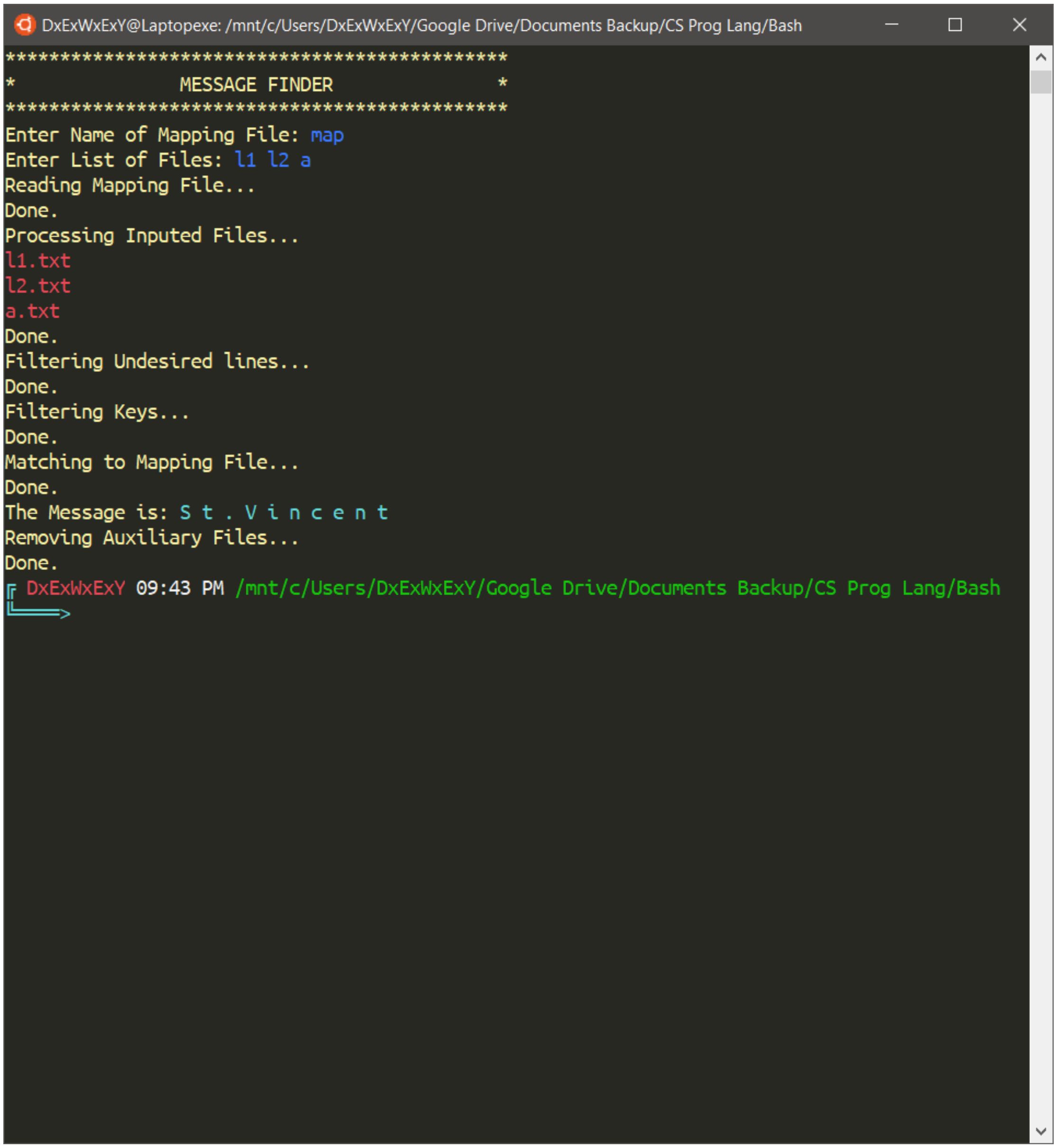
```bash
#!/bin/bash
clear
echo "$(tput setaf 3)**********************************************"
echo "*               MESSAGE FINDER               *"
echo "**********************************************"
# Prompts user for name of mapping file and the
# files to be examined, assuming the file is .txt
read -p "$(tput setaf 3)Enter Name of Mapping File: $(tput setaf 4)" mapFile
read -p "$(tput setaf 3)Enter List of Files: $(tput setaf 4)" files
# creates an array with the file names
declare -a fileList
readarray fileList <<< $files
# sorts the mapping file by the numbers in the last column
echo "$(tput setaf 3)Reading Mapping File..."
sort -k3,3n "$mapFile.txt" > _AUXstmp.txt
echo "Done."
# creates an auxiliary file and appends the lines
# with "error" as well as the line below found in
# the input files
echo "Processing Inputed Files..."
touch _AUXf.txt
for file in $fileList; do
        echo "$(tput setaf 1)$file.txt"
        grep 'error' -A 1 "$file.txt" >> _AUXf.txt
done
echo "$(tput setaf 3)Done."
# deletes the lines with "syntax, server, memory,"
# and the line below, then it deletes the lines that
# contain error
echo "Filtering Undesired lines..."
sed '/syntax/,+1d' -i _AUXf.txt
sed '/server/,+1d' -i _AUXf.txt
sed '/memory/,+1d' -i _AUXf.txt
sed '/error/d' -i _AUXf.txt
echo "Done."
# retrieves all the words of length 2 at the beginning
# of each line in the first filter file
echo "Filtering Findings..."
grep -o -w -E '^[a-z]{2}' _AUXf.txt > _AUXf2.txt
echo "Done."
# if a line in _AUXstmp contains a word found in _AUXf2,
# then it will write the line from _AUXstmp to _AUXmsg,
# thus filtering false positives length 2 words found
echo "Matching to Mapping File..."
grep -Fwf _AUXf2.txt _AUXstmp.txt > _AUXmsg.txt
# this extracts the all the letters in the third column
# into _AUXword, then it's concatenated into a string
cut -c 4 _AUXmsg.txt > _AUXword.txt
declare -a letters
readarray -t letters <<< $( cat _AUXword.txt )
word="The Message is:$(tput setaf 6)"
for letter in $letters; do
        word="$word $letter"
done
echo "Done."
# prints the message found and removes all the auxiliary files created
echo $word
echo "$(tput setaf 3)Removing Auxiliary Files..."
rm _AUX*.txt
echo "Done."
```

Diego Rodriguez

Although Bash is very powerful, if the programmer is not familiarized with its functions and features, it tricky for newcomers. In my case, I wasn't aware that some programs could not do everything that I intended to, for example I tried to use grep to delete lines containing the forbidden words and the lines below, but all I could do was to invert the search using -v and creating a new file every time I wanted to delete each of the forbidden words, and thanks to a Google search I found out about sed deletion function and how it could delete lines straight from the file, rather than creating a new file each time. Due to my limited knowledge, I came up with a clever way to concatenate the found letters into a string, since I couldn't find a way to do it directly, first I extracted all the letters in the 4th column into a file, then I read the file into an array (readarray reads every element separated by whitespace into an index, not every new line), then I concatenated the letters into a string using a for loop.


DxExWxExY@Laptopexe: /mnt/c/Users/DxExWxExY/Google Drive/Documents Backup/CS Prog Lang/Bash

```
**********************************************
*               MESSAGE FINDER               *
**********************************************
Enter Name of Mapping File: map
Enter List of Files: l1 l2 a
Reading Mapping File...
Done.
Processing Inputed Files...
l1.txt
l2.txt
a.txt
Done.
Filtering Undesired lines...
Done.
Filtering Keys...
Done.
Matching to Mapping File...
Done.
The Message is: S t . V i n c e n t
Removing Auxiliary Files...
Done.
╓ DxExWxExY 09:43 PM /mnt/c/Users/DxExWxExY/Google Drive/Documents Backup/CS Prog Lang/Bash
╙═════>
```