

Assignment 2 ,4 and 5

[illegible]

```
varun@DESKTOP-9EIE2GC:/n X + v
#!/bin/bash

function find_files()
{
    var= df -h | awk '{if($5<10) print $1}'
    echo $var
}

find_file$
```

"a4.sh" 9L, 117B

39°C Mostly sunny

Search

ENG IN 03:17 PM 19-04-2023

38°C Light rain Search ENG IN 03:48 PM 19-04-2023

```

"06_reverse.sh" 18L, 406B

```


38°C Light rain Search ENG IN 04:03 PM 19-04-2023

38°C Light rain Search ENG IN 04:03 PM 19-04-2023


```
varun@DESKTOP-9EIE2GC: /n X + v
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ vi 14_file_upper_lower.sh
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ ls
01_pattern.sh 03_real_add.sh 05_largest.sh 07_delete_emptylines.sh 09-fibonacci.sh 11_chess_board.sh 14_file_upper_lower.sh
02_pattern.sh 04_calculator.sh 06_reverse.sh 08_operator_dependent.sh 10_string_length.sh 12_sorting.sh empty.txt
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ ./14_file_upper_lower.sh
Before renaming:
.:
01_pattern.sh 03_real_add.sh 05_largest.sh 07_delete_emptylines.sh 09-fibonacci.sh 11_chess_board.sh 14_file_upper_lower.sh
02_pattern.sh 04_calculator.sh 06_reverse.sh 08_operator_dependent.sh 10_string_length.sh 12_sorting.sh empty.txt
Renaming files/directories to lowercase:
Renaming files/directories to uppercase:
mv: '01_pattern.sh' and '01_PATTERN.SH' are the same file
mv: '02_pattern.sh' and '02_PATTERN.SH' are the same file
mv: '03_real_add.sh' and '03_REAL_ADD.SH' are the same file
mv: '04_calculator.sh' and '04_CALCULATOR.SH' are the same file
mv: '05_largest.sh' and '05_LARGEST.SH' are the same file
mv: '06_reverse.sh' and '06_REVERSE.SH' are the same file
mv: '07_delete_emptylines.sh' and '07_DELETE_EMPTYLINES.SH' are the same file
mv: '08_operator_dependent.sh' and '08_OPERATOR_DEPENDENT.SH' are the same file
mv: '09-fibonacci.sh' and '09-FIBONACCI.SH' are the same file
mv: '10_string_length.sh' and '10_STRING_LENGTH.SH' are the same file
mv: '11_chess_board.sh' and '11_CHESS_BOARD.SH' are the same file
mv: '12_sorting.sh' and '12_SORTING.SH' are the same file
mv: '14_file_upper_lower.sh' and '14_FILE_UPPER_LOWER.SH' are the same file
mv: 'empty.txt' and 'EMPTY.TXT' are the same file
After renaming:
.:
01_pattern.sh 03_real_add.sh 05_largest.sh 07_delete_emptylines.sh 09-fibonacci.sh 11_chess_board.sh 14_file_upper_lower.sh
02_pattern.sh 04_calculator.sh 06_reverse.sh 08_operator_dependent.sh 10_string_length.sh 12_sorting.sh empty.txt
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ ls
01_pattern.sh 03_real_add.sh 05_largest.sh 07_delete_emptylines.sh 09-fibonacci.sh 11_chess_board.sh 14_file_upper_lower.sh
02_pattern.sh 04_calculator.sh 06_reverse.sh 08_operator_dependent.sh 10_string_length.sh 12_sorting.sh empty.txt
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ |
```

```
varun@DESKTOP-9EIE2GC: /n X + v
# Function to rename a file/directory with lower case letters
rename_lowercase() {
    local old_name=$1
    local new_name=$(echo "$old_name" | tr '[:upper:]' '[:lower:]')
    if [[ "$old_name" != "$new_name" ]]; then
        mv -v "$old_name" "$new_name"
    fi
}

# Function to rename a file/directory with upper case letters
rename_uppercase() {
    local old_name=$1
    local new_name=$(echo "$old_name" | tr '[:lower:]' '[:upper:]')
    if [[ "$old_name" != "$new_name" ]]; then
        mv -v "$old_name" "$new_name"
    fi
}

# Main program
echo "Before renaming:"
ls -R
echo "Renaming files/directories to lowercase:"
for item in *; do
    if [[ -f "$item" ]]; then
        rename_lowercase "$item"
    elif [[ -d "$item" ]]; then
        cd "$item"
        for subitem in *; do
            rename_lowercase "$subitem"
        done
        cd ..
        rename_lowercase "$item"
    fi
done
echo "Renaming files/directories to uppercase:"
for item in *; do
    if [[ -f "$item" ]]; then
        rename_uppercase "$item"
    elif [[ -d "$item" ]]; then
        cd "$item"
        for subitem in *; do
            rename_uppercase "$subitem"
        done
        cd ..
        rename_uppercase "$item"
    fi
done
echo "After renaming:"
ls -R
```

```
varun@DESKTOP-9EIE2GC: /n  X + v
#!/bin/bash
# prompt user for number of passwords to generate
echo "How many random passwords would you like to generate?"
read num_passwords
# loop to generate passwords
for ((i=1;i<=$num_passwords;i++)); do
    password=$(openssl rand -base64 6 | tr -d "+/*")
    echo "Password $i: $password"
done

```

"20_random_password.sh" 9L, 296B

```
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ vi 20_random_password
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ rm 20_random_password
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ vi 20_random_password.sh
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ ./20_random_password.sh
How many random passwords would you like to generate?
5
Password 1: WgrFWHG
Password 2: 70fs0ngK
Password 3: 901ZEAzv
Password 4: yzF3z8LE
Password 5: InUCtM1
varun@DESKTOP-9EIE2GC: /mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ |

```



```
varun@DESKTOP-9EIE2GC: /n  +  v
varun@DESKTOP-9EIE2GC:/mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ vi 18_largest_uname.sh
varun@DESKTOP-9EIE2GC:/mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ ./18_largest_uname.sh
The user with longest name is systemd-timesync with length 16
The user with shortest name is lp with length 2
varun@DESKTOP-9EIE2GC:/mnt/c/Users/Varun/Desktop/INFOBELL/CDAC-Batch-2-Training/Varun Garade/19-04-2023/practice$ |
```

```
varun@DESKTOP-9EIE2GC: /n  +  v
#!/bin/bash
# get the usernames and their lengths from the passwd file using awk
usernames=$(awk -F: '{ print $1, length($1) }' /etc/passwd)
# get the longest and shortest usernames using sort and head/tail
longest=$(echo "$usernames" | sort -k2 -rn | head -n1)
shortest=$(echo "$usernames" | sort -k2 -rn | tail -n1)
# extract the username and length from the output
longest_username=$(echo "$longest" | awk '{ print $1 }')
longest_length=$(echo "$longest" | awk '{ print $2 }')
shortest_username=$(echo "$shortest" | awk '{ print $1 }')
shortest_length=$(echo "$shortest" | awk '{ print $2 }')
# print the result
echo "The user with longest name is $longest_username with length $longest_length"
echo "The user with shortest name is $shortest_username with length $shortest_length"
"
```

"18_largest_uname.sh" 14L, 785B