Q1. Write a bash script to print the directory tree (iterate over sub-directories as well) from a directory given as a parameter to the script. While printing the name of the files, remove the extension of the file. Also, count the number of files and directories.

The file **mytree.sh** is as follows.

```
#!/bin/bash

BRANCH_SYMBOL="└──"
DIR_COUNT=0
FILE_COUNT=0

# returns a string that represents the name of the file without extension
# $1: file name
remove_extension() {
    FILE_NAME=$(echo $1)
    REV_FILE_NAME=$(echo $1 | rev)

    for (( i=0; i<${#REV_FILE_NAME}; i++ ))
    do
        if [ "${REV_FILE_NAME:$i:1}" = "." ]
            then
                INDEX=$((i + 1))
                FILE_NAME=$(echo "${REV_FILE_NAME:$INDEX:${#FILE_NAME}}" | rev)
                break
        fi
    done

    echo $FILE_NAME
}

# prints the name of the directory or file
# $1: number of offset
# $2: name of file(not full path)
# $3 : 1 if file otherwise 0
print() {
    SPACE_LEN=$(( $1 * 4 ))
    EMPTY_STRING=$(printf %${SPACE_LEN}s)
    LINE="${EMPTY_STRING}${BRANCH_SYMBOL}"
    FILE_NAME="$2"
    if [ $3 -eq 1 ]
        then
            FILE_NAME=$(remove_extension $FILE_NAME)
    fi
    echo "${LINE}${FILE_NAME}"
}

# recursively traverses the directory and prints the name of directory and file.
also calculates the total number
# of directories and files in global variables defined at the beginning of file
# $1: path to directory
# $2: number of offset
dfs() {
```

```
    for i in $(ls $1)
    do
        if [ -d "${1}/${i}" ]
            then
                (( DIR_COUNT++ ))
                print $2 "${i}" 0
                dfs "${1}/${i}" $(( $2 + 1 ))
            else
                (( FILE_COUNT++ ))
                print $2 $i 1
        fi
    done
}

# call the dfs function and pass the name of the directory given as arg. second
argument represents the indentation level
dfs $1 0

# print the count of directories and files
echo
echo "${DIR_COUNT} directories, ${FILE_COUNT} files"
```

Following command is used to run the program.

```
./mytree.sh ../../../assignments
```

The output of the above script is as follows.

```
        └──input
        └──main
        └──matrix
        └──matrix
        └──screenshot
            └──1
    └──4
        └──first
        └──first
        └──main
        └──main
        └──main-static
        └──mylib
        └──mylib
        └──second
        └──second
    └──MIT2021117-PPR-A2
    └──PPR-Assignment-2
└──3
    └──1
        └──1-1
        └──1-2
        └──1-3
        └──1-4
```

```
        └──1-2
        └──1-3
        └──1-4
    └──2
        └──2-1
        └──2-2
        └──2-3
        └──2-4
        └──2-5
        └──2-6
        └──cpu-bound
        └──cpu-bound
        └──io-bound
        └──io-bound
    └──3
        └──a
        └──notes
        └──output
        └──q3
    └──4
        └──4-1
        └──4-2
        └──a
        └──q4
```

```
        └──4-2
        └──a
        └──q4
    └──5
        └──a
        └──output
        └──q5
    └──6
        └──a
        └──management
        └──q6
    └──7
        └──a
        └──client
        └──output
        └──server
        └──server
    └──8
        └──inputs
            └──input
        └──input
        └──output-1
        └──output-2
        └──output-3
```


└── sort1
└── sort1
└── xsort
└── xsort
└── 9
└── notes
└── row-mul
└── row-mul
└── row-mul
└── serial-mul
└── serial-mul
└── serial-mul
└── notes
└── PPR-assignment-3
└── questions
└── 5
└── 1
└── mytree
└── 2
└── assig5-shell
└── PPR-assignment-5.md
23 directories, 83 files
@1 👀