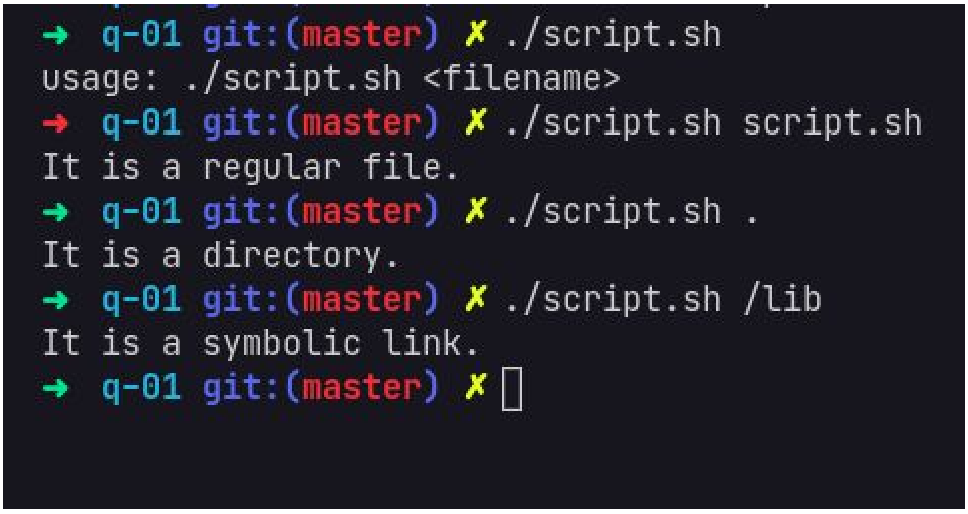
OS LAB 4

SAHIL BONDRE: U18CO021

1. Write a shell script which takes filename as argument and checks whether file is regular file,directory,block special file,character special file,named pipe,symbolic link,socket,device file etc.

```
if [ $# -ne 1 ]; then
  echo "usage: ./script.sh <filename>"
  exit 2
fi
filename=$1

ls -ld $filename >temp.txt
file=temp.txt
res=$(cat "$file")
type=$(echo $res | cut -c1-1)
case $type in
"-") echo "It is a regular file." ;;
"d") echo "It is a directory." ;;
"c") echo "It is a character device file" ;;
"l") echo "It is a symbolic link." ;;
"s") echo "It is a local socket file." ;;
"b") echo "It is a block device file." ;;
"p") echo "It is a named pipe." ;;
*) echo "It is not a valid type." ;;
esac

rm temp.txt
```

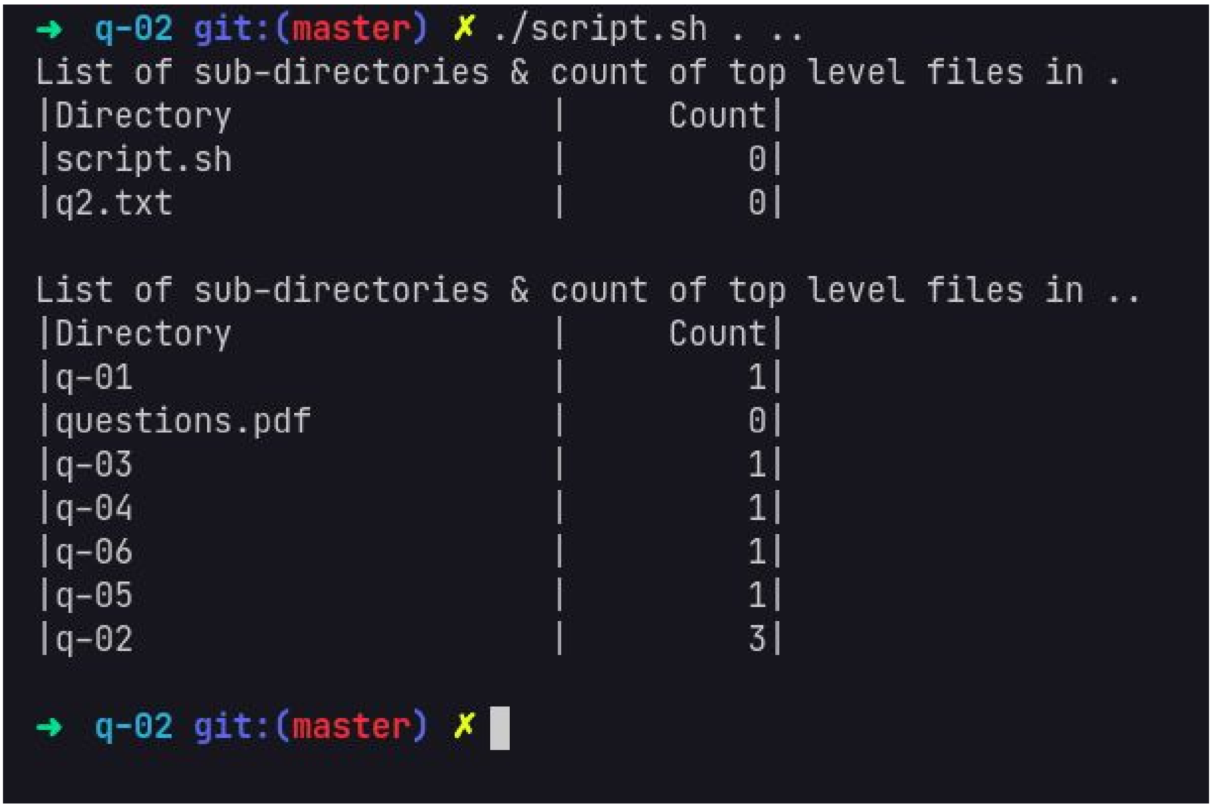


```
➜  q-01 git:(master) ✗ ./script.sh
usage: ./script.sh <filename>
➜  q-01 git:(master) ✗ ./script.sh script.sh
It is a regular file.
➜  q-01 git:(master) ✗ ./script.sh .
It is a directory.
➜  q-01 git:(master) ✗ ./script.sh /lib
It is a symbolic link.
➜  q-01 git:(master) ✗
```

.

2. Write a shell script which will take file name as argument and check whether the file name is a dir or not and then proceed further only if it is a dir, else give usage message. The script should then print in the tabular format, name of each sub-dir (within the argument dir) and a count of the number of top level files in that sub-dir. Modify the program to work with multiple number of arguments, too

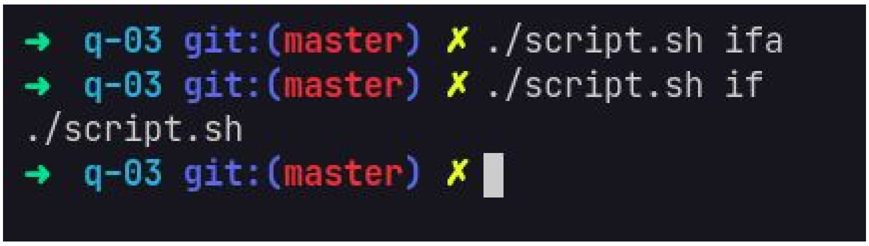
```
if [ $# -eq 0 ]; then
  echo "usage: ./script.sh [<dirs>,..]"
  exit 2
fi


for filename in "$@"; do
  if [ -d "$filename" ]; then
    find $filename -depth -maxdepth 1 >q2.txt
    [ -f "count.txt" ] && rm count.txt
    [ -f "final.txt" ] && rm final.txt
    echo "List of sub-directories & count of top level files in $filename"
    echo "Directory,Count" >>count.txt
    while read line; do
      if [[ $filename != $line ]]; then
        count=0
        for entry in "$line"/*; do
          [ -f "$entry" ] && count=$(($count + 1))
        done
        file=$(echo basename $line)
        new="$file,$count"
        $new >>count.txt
      fi
    done <q2.txt

    while IFS=, read -r a b; do echo "$a $b" >>final.txt; done <count.txt
    awk '{printf "|%-25s|%10s|\n",$1,$2}' final.txt
    echo ""
  else
    echo "$fileName is not a directory"
  fi
done

rm *.txt
```



```
➜  q-02 git:(master) ✗ ./script.sh . ..
List of sub-directories & count of top level files in .
|Directory                |     Count|
|script.sh                |         0|
|q2.txt                   |         0|

List of sub-directories & count of top level files in ..
|Directory                |     Count|
|q-01                     |         1|
|questions.pdf            |         0|
|q-03                     |         1|
|q-04                     |         1|
|q-06                     |         1|
|q-05                     |         1|
|q-02                     |         3|

➜  q-02 git:(master) ✗ 
```

3. Write a script that will search for a specific word in all the files in the current dictionary and then prompt with the file name in which word is found

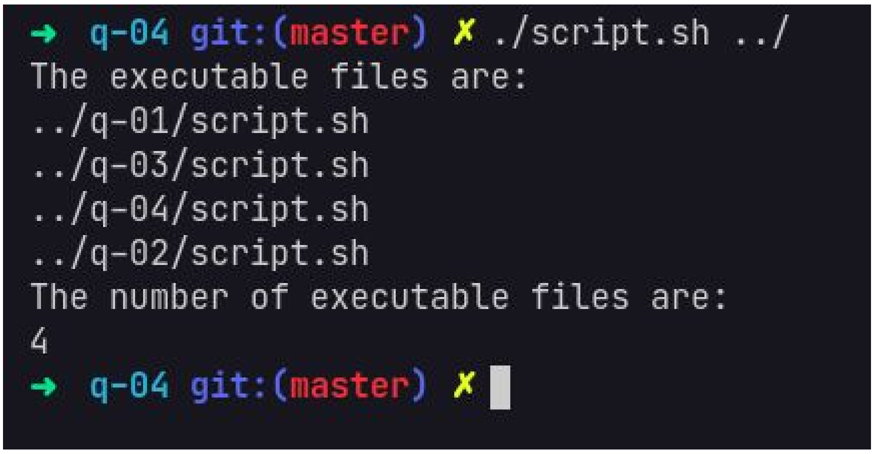
```
if [ $# -ne 1 ]; then
  echo "usage: ./script.sh <word>"
  exit 2
fi
files=$(find . -type f)

word=$1

for file in $files; do
  res=$(grep -w $word $file)
  if [ -n "$res" ]; then
    echo $file
  fi
done
```



```
➜  q-03 git:(master) ✗ ./script.sh ifa
➜  q-03 git:(master) ✗ ./script.sh if
./script.sh
➜  q-03 git:(master) ✗ 
```

4. Write a script to print only the number of executable file in each sub-dir of the argument directory specified.

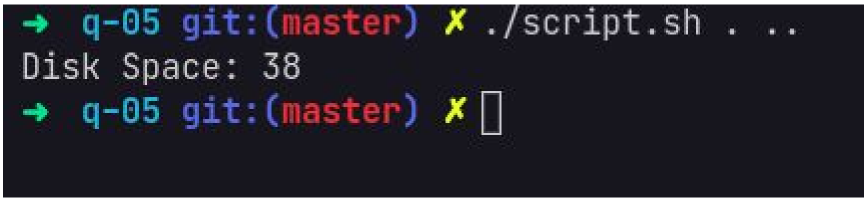
```
if [ $# -ne 1 ]; then
  echo "usage: ./script.sh <dir>"
  exit 2
fi
name=$1

echo "The executable files are:"
find $name -executable -type f

echo "The number of executable files are:"
find $name -executable -type f | wc -l
```



```
➜  q-04 git:(master) ✗ ./script.sh ../
The executable files are:
../q-01/script.sh
../q-03/script.sh
../q-04/script.sh
../q-02/script.sh
The number of executable files are:
4
➜  q-04 git:(master) ✗ 
```

5. Write a non-interactive script that takes in any no. of directory name as argument and calculates total no. of blocks of disk space occupied by the ordinary files in all the directories

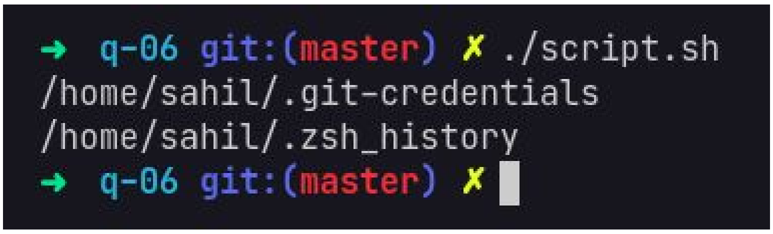
```
count=0
for arg in "$@"; do
  if [ -d $arg ]; then
    res=$(find $arg -type f)
    for file in $res; do
      ls -ld $file >q1.txt
      fileNew=q1.txt
      res=$(cat "$fileNew")
      type=$(echo $res | cut -c1-1)
      if [ $type == "-" ]; then

        size=$(du -sc $file | tail -n 1 | cut -c1-1)
        count=$(($count + $size))
      fi
    done
  fi
done

rm q1.txt
echo "Disk Space: $count"
```



```
➜  q-05 git:(master) ✗ ./script.sh . ..
Disk Space: 38
➜  q-05 git:(master) ✗ ▯
```

6. Write a shell script file named exercise2.sh that makes a list of files in your home directory that were changed less than 24 hours ago, but leave out directories.

```
find ~ -mtime -1 -type f -maxdepth 1 2> /dev/null
```



```
➜  q-06 git:(master) ✗ ./script.sh
/home/sahil/.git-credentials
/home/sahil/.zsh_history
➜  q-06 git:(master) ✗ █
```