1. **Print the number of directories.**

   Straightforward solution:

```
#!/bin/bash

n=0
for file in *
do
    if [[ -d $file ]]
    then
        n=$((n+1))
    fi
done
echo Number of directories: $n
```

   Different solution using pipes:

```
#!/bin/bash

echo Number of directories: `ls -l | grep ^d | wc -l`
```

2. **Print the names of all TXT files in the current directory (including its subdirectories) that contain the word ”bone”.**

```
#!/bin/bash

str=bone
echo The following text files contain the word "$str":
find . -type f -name "*.txt" -exec grep -l "$str" {} ";"
```

3. **Print the total size of TXT files in the current directory (including its subdirectories).**

```
#!/bin/bash

bytes=0
OIFS=$IFS
IFS=$'\n'
for file in `find . -type f -name "*.txt"`
do
    bytes=$((bytes + $(wc -c < "$file")))
done
IFS=$OIFS
echo Total size of text files: $bytes
```

   Different solution using process substitution:

```
#!/bin/bash

bytes=0
while IFS= read -r -d '' file; do
    bytes=$((bytes + $(wc -c < "$file")))
done < <(find . -type f -name "*.txt" -print0)
echo Total size of text files: $bytes
```

Another variation:

```
#!/bin/bash

k=0
find . -type f -name "*.txt" -exec wc -c {} ";" | cut -d ’ ’ -f1 | sh -c ’
    bytes=0
    while read line
    do
        bytes=$((bytes+line))
    done
    echo Total size of text files: $bytes
’
```

4. **Find all TXT files whose names contain ”2017” and replace it with ”2018”.**

```
#!/bin/bash

n=0
OIFS=$IFS
IFS=$’\n’
for file in ‘find . -type f -name "*2017*.txt"‘
do
    mv "${file}" "${file//2017/2018}"
    n=$((n+1))
done
IFS=$OIFS
echo $n files renamed.
```

5. **Standardize the names of all files in the current directory (nonrecursively): change all filenames to lowercase, convert spaces and hyphens to underscore.**

```
#!/ bin/ bash

n=0
for file in *
do
    if [[ $file = $0 ]]
    then
        continue
    fi
    newname=‘echo "$file" | tr "[A-Z]" "[a-z]" | tr "[ \-]" "_"‘
    if [[ $newname != $file ]]
    then
        n=$((n+1))
        echo "$n: $file --> $newname"
        mv -i "$file" "$newname"
    fi
done
```

Another variation, if GNU extensions are supported:

```
#!/ bin/ bash

n=0
for file in *
do
    if [[ $file = $0 ]]
    then
        continue
    fi
    newname=‘echo "$file" | sed "s/\(.*\)/\L\1/ ; s/[ -]/_/g"‘
    if [[ $newname != $file ]]
    then
        n=$((n+1))
        echo "$n: $file --> $newname"
        mv -i "$file" "$newname"
    fi
done
```

6. **Number all files in the current directory (nonrecursively) by their modification date from oldest to latest. Append the number to the beginning of the filename.**

```
#!/ bin/ bash

n=0
for file in ‘ls -tr‘
do
    if [[ $file = $0 ]]
    then
        continue
    fi
    n=$((n+1))
    mv -i "$file" "${n}_${file}"
done
```

Similar solution using pipes:

```
#!/ bin/ bash

n=0
ls -tr | sed "/$0/d" | while read file
do
    n=$((n+1))
    mv -i "$file" "${n}_${file}"
done
```

7. **Modify CODE.TXT so that each line is terminated with a semicolon. Ignore blank lines and lines that are already terminated.**

```
#!/bin/bash

sed "/^\s*$/b ; /;\s*$/b ; s/\s*$/;/g" -i CODE.TXT
```

Different solution:

```
#!/bin/bash

file=CODE.TXT
tempfile=CODE.TXT2
while read line
do
    if echo $line | grep -q "^\s*$\|;\s*$"
    then
        echo $line
    else
        echo "${line//[$’\t\r\n’]/;}"
    fi
done < $file > $tempfile
mv $tempfile $file
```