Week 5

Practice Assignment

Problem 1

What is the expected behaviour of the following code?

```
#!/bin/bash

for file in `ls *.sh`; do
    bash ./$file
done
```

(1) Runs all the scripts in the current working directory once.

(2) Runs all the scripts in the current directory, but returns error for the scripts that expects operands.

(3) May raise an error for each script in current directory as the file type is not specified.

(4) Will enter into an infinite loop.

Answer

(2) and (4)

Problem 2

What is the expected behaviour of the following bash script?

```
for file in `find . -maxdepth 1 -name '*.txt'`; do
        echo $(basename $file)
done
```

(1) Prints the file names of all the `.txt` files in the current directory non recursively.

(2) Prints the file names of all tthe `.txt` files in the current directory recursively.

(3) Prints the file names of all tthe `.txt` files in the current directory and the directories directly into current directory.

Answer

(1)

Problem 3

The following bash script is written to count the number of empty files in the present working direcory. Due to some error in the code, it does not function as expected. Which of the following changes should be made to complete the required task?

```
c=0
for file in ./; do
    if ! [ -z `file file | grep 'empty'` ]
    then
        c++;
    fi
done;
```

echo $c

(1)

```
c=0
for file in ./*; do
    if ! [ -z `file file | grep 'empty'` ]
    then
        c++;
    fi
done;
```

(2)

```
c=0
for file in ./*; do
    if ! [ -z "`file file | grep 'empty'`" ]
    then
        ((c++));
    fi
done;
```

(3)

```
c=0
for file in ./*; do
    if ! [ -z "`file $file | grep 'empty'`" ]
    then
        ((c++));
    fi
done;
echo $c
```

(4)

```
c=0
for file in ./*; do
    if ! [ -z `file $file | grep 'empty'` ]
    then
        ((c++));
    fi
done;
echo $c
```

Answer

(3)

Problem 4(Let it be MSQ even if only one answer is correct)

Which of the following options correctly specifies the usage and output of the script below? [MSQ]

```
function change()
{
    if [ $1 -eq "0" ]; then
        echo "A"
    elif [ $1 -eq "1" ]; then
        echo "B"
    elif [ $1 -eq "2" ]; then
        echo "C"
    elif [ $1 -eq "3" ]; then
        echo "D"
    elif [ $1 -eq "4" ]; then
        echo "E"
    elif [ $1 -eq "5" ]; then
        echo "F"
    elif [ $1 -eq "6" ]; then
```

```
        echo "G"
    elif [ $1 -eq "7" ]; then
        echo "H"
    elif [ $1 -eq "8" ]; then
        echo "I"
    elif [ $1 -eq "9" ]; then
        echo "J"
    else
        echo "X"
    fi
}
```

(1) Use: `change(5)` | Output: `F`

(2) Use: `change{5}` | Output: `F`

(3) Use: `change 5` | Output: `F`

(4) Use: `$( change 5 )` | Output: `F`

Answer

(4)

Question 5

```
#!/bin/bash

for myd in `find . -type d`
do
    for n in `ls $myd`
    do
        t=`file $myd/$n|grep "shell script"`;
        if [ -n "$t" ]
        then
            echo $n
        fi
    done
done
```

Which of the following commands will give the same output as the above script? [MCQ]

A. `ls -R | xargs file | grep shell | grep -o "^.*\.sh"`
B. `ls -R | file | grep shell | grep -o "^.*\.sh"`
C. `ls -R | xargs file | grep -v shell | grep -o "^.*\.sh"`
D. `ls -R | xargs file | grep shell | grep  "^.*\.sh"`

Answer:

(A)

Problem 6

Consider a file currently opened in the vi editor with the below text.

```
Line1
Line2
Line3
Line4
...
...
Line97
Line98
Line99
Line100
```

Given below in each option is a command(in vi editor) and it function. Choose the correct pairs from the options below. [MSQ]

(1) `3,$s/Line/line/g`, Performs 98 substitutions

(2) `%s/line/Line/g`, Performs 100 substitutions

(3) `%s/blah/Line/g`, E486: Pattern not found: blah

(4) `%/Line/Line`, Performs 100 substitutions

Answer

(1) `3,$s/Line/line/g` | Performs 98 substitutions

(2) `%s/line/Line/g` | Performs 100 substitutions

(3) `%s/blah/Line/g` | E486: Pattern not found: blah