Week 5

Graded Assignment 2 (compulsory)

Problem 1

Given below the contents of the bash script `file.sh`. [MSQ]

```
#!/bin/bash

sum=0
for (( i=1 ; i<$1 ; i++ )) ; do
    if [ $((i%2)) -eq 1 ]; then
        sum=$((sum+i))
    fi
done
echo $sum
```

Choose the option(s) for which the expected output matches the script usage.

(1) Usage: `./file.sh 45` | Output: 484

(2) Usage: `./file.sh 45` | Output: 1035

(3) Usage: `./file.sh 57` | Output: 1653

(4) Usage: `./file.sh 45` | Output: 784

Answer

(1) and (4)

Problem 2

What is the expected output of the following bash script? [MCQ]

```
#!/bin/bash
for i in `ls`; do
        echo $(pwd)/$i
done
```

(1) Prints the file names only of all the files in the current directory.

(2) Prints the relative file paths of all the files in the current directory.

(3) Prints the absolute file paths of all the files in the current directory.

(4) None of the above

Answer

(3)

Problem 3

`number` is a shell variable. Which of the following condition checks if `number` is divisible by 3 or not? [MSQ]

(1) `[ $(number%3) -eq 0 ]`

(2) `[ $((number%3)) -eq 0 ]`

(3) `[ $(($number%3)) -eq 0 ]`

(4) `[ number%3 -eq 0 ]`

Answer

(2) `[ $((number%3)) -eq 0 ]`

(3) `[ $(($number%3)) -eq 0 ]`

Problem 4

What of the following scripts will combine the text in all the `.txt` files in the current working directory to a single file? [MCQ]

(1)

```
for file in `ls -l *.txt`; do
    cat $file > ../allfiles.txt
done
```

(2)

```
for file in `ls *.txt`; do
    cat $file > ../allfiles.txt
done
```

(3)

```
for file in `ls *.txt`; do
    cat $file >> ../allfiles.txt
done
```

(4)

```
for file in `ls -l *.txt`; do
    cat $file >> ../allfiles.txt
done
```

Answer

(2)

Problem 5

When the command `ls -l` is run on the current directory, the output is.

```
-rw-r--r--  1 user  group   0 Nov 30 11:08 rand1.txt
-rw-r--r--  1 user  group   0 Nov 29 11:08 rand2.txt
-rw-r--r--  1 user  group   0 Nov 29 11:08 rand3.md
-rw-r--r--  1 user  group   0 Nov 28 11:08 rand4.awk
-rwxr-xr-x 1 user  group   0 Nov 10 14:03 script.sh
-rwxr-xr-x 1 user  group  1 Nov 30 20:44 test.sh
```

What is the correct output on running the below bash script? [MCQ]

```
for file in `ls`; do
        details=`ls -l $file`
        echo $file: ${details:0:10}
done
```

(1)

```
rand1.txt: details:0:10
rand2.txt: details:0:10
rand3.md: details:0:10
rand4.awk: details:0:10
script.sh: details:0:10
test.sh: details:0:10
```

(2)

```
rand1.txt: user
rand2.txt: user
rand3.md: user
rand4.awk: user
script.sh: user
test.sh: user
```

(3)

```
rand1.txt: group
rand2.txt: group
rand3.md: group
rand4.awk: group
script.sh: group
test.sh: group
```

(4)

```
rand1.txt: -rw-r--r--
rand2.txt: -rw-r--r--
rand3.md: -rw-r--r--
rand4.awk: -rw-r--r--
script.sh: -rwxr-xr-x
test.sh: -rwxr-xr-x
```

Answer

(4)


```
rand1.txt: -rw-r--r--
rand2.txt: -rw-r--r--
rand3.md: -rw-r--r--
rand4.awk: -rw-r--r--
script.sh: -rwxr-xr-x
test.sh: -rwxr-xr-x
```

Problem 6

John wants to read two numbers from the user and print the sum total in USD(with a $ sign before the number). Which of the following bash scripts can do this? [MCQ]

(1)

```
read "Enter the first value:" a
read "Enter the second value:" b

echo $$((a+b))
```

(2)

```
read "Enter the first value:" a
read "Enter the second value:" b

echo \$$((a+b))
```

(3)

```
echo "Enter the first value:" && read  a
echo "Enter the second value:" && read b

echo $$((a+b))
```

(4)

```
echo "Enter the first value:" && read  a
echo "Enter the second value:" && read b

echo \$$((a+b))
```

Answer

(4)

Problem 7

Consider the following bash script stored as `function.sh`.

```
read var
function operate()
{
    temp=$1
    temp=${temp%.*}
    echo $temp
}

echo $( operate $var )
```

If the script is run from the terminal as `./function.sh` and the then number `45.53` is entered on the terminal, what is the expected output? [NAT]

Answer

45

Question 8

```
for i in file_{A..D}{1..3}
do
    n=$i.txt
    touch $n
done
```

How many files will be created when the above script is executed? [NAT]

Answer

12

Question 9

Select all commands/scripts that will count the number of words in the file named `paragraph`. [MSQ]

(a) `wc -w paragraph`
(b) `wc -w < paragraph`
(c) `sum=0; for i in $(cat paragraph); do ((sum++)); done; echo $sum`
(d)

```
sum=0;
while read line; do
  for word in $line; do
    if [[ $line != "" ]]; then
      ((sum++))
    fi
  done
  shift;
done < paragraph
echo $sum
```

Answer

(a), (b), (c) and (d)