Week 6 - Graded Assignment

Problem 1

`file1.txt` is a file containing some text . The command `awk '{print}' file1.txt` is equivalent to which of the following commands, irrespective of the data in `file1.txt`. [MSQ]

(1) cat file1.txt

(2) cut -f 1- file1.txt

(3) head file1.txt

(4) grep "" file1.txt

Answer

(1), (2) and (4)

Problem 2

What is the output of the below command?

```
for i in {1..9}; do echo $i; done | awk 'BEGIN{value=0;}{value += $1*$1;}END{print value;}'
```

[MCQ]

(1) 45

(2) 204

(3) 37

(4) 285

Answer

(4) 285

Problem 3

What is the output of the following bash script?
Note that before executing the below script the file `numbers.txt` is empty. [MCQ]

```
for i in {1..10}; do
  for j in {1..10}; do
    echo $i, $j >> numbers.txt
  done
done

awk 'BEGIN{FS=",";value=0;}{value += $1*$2;}END{print value;}' numbers.txt
```

(1) 3025

(2) 2025

(3) 45

(4) 55

Answer

(1) 3025

Problem 4

In which of the following commands output, the 1st and 2nd column of input will be inverted? In the input, the 1st and 2nd columns are separated by a comma, and in the output too the columns should be separated by a comma.[MSQ]

(1) `awk 'BEGIN{FS=",";}{print$2,$1}'`

(2) `awk '{print$2","$1}'`

(3) `awk 'BEGIN{FS=",";}{print$2","$1}'`

(4) `awk '{print$2,$1}'`

Answer

(3)

Problem 5

The purpose of the following command is to

```
ls -l | egrep -v "^d" | awk '{sum += $5}; END{print sum}'
```

(1) Calculate the number of characters in all the files in the current directory.

(2) Calculate the space occupied by all the files in the current directory in bytes.

(3) Calculate the sum of creation dates of all the files in the current directory.

(4) Calculate the number of lines in all the files in the current directory.

Answer

(2)

Problem 6

```
$ cat employee_details.txt
```

The output of the above command is as follows;

```
A1998001,Ram Kumar,10,Male
B2000002,Sanjay Narayan,7,Male
B2000003,Srishti Rai,10,Female
E1997001,Manoj Pillai,12,Male
G1998001,Preeti Suresh,9,Female
G1999001,Leela L G,16,Female
```

What is the output of the command displayed below?

```
$ awk 'NR==3, NR==5 {print NR,$1}' employee_details.txt
```

[MCQ]

(1)

```
3 B2000003
5 G1998001
```

(2)

```
3 B2000003,Srishti Rai,10,Female
5 G1998001,Preeti Suresh,9,Female
```

(3)

```
3 B2000003
4 E1997001
5 G1998001
```

(4)

```
3 B2000003,Srishti
4 E1997001,Manoj
5 G1998001,Preeti
```

Answer

(4)

Problem 7

Which of the following commands can be used to calculate the number of lines in the file `employee_details.txt`. [MSQ]

(1)

```
$ awk 'END { print NR }' employee_details.txt
```

(2)

```
$ awk 'END {count}' employee_details.txt
```

(3)

```
$ awk '{++value;}END{print value;}' employee_details.txt
```

(4)

```
$ awk 'BEGIN{FS=",";n=0}{n++;print $n}' employee_details.txt
```

Answer

(1) and (3)

Problem 8

```
$ echo To be or not to be | tr " " "\n"
```

Which of the following awk commands will give same output as the command above? **(MCQ)**

Note: `tr " " "\n"` will replace each space by a newline character. i.e. each word in the string "To be or not to be" will be printed on a new line by the command `echo To be or not to be | tr " " "\n"`

(1)

```
$ awk 'BEGIN{sentence="To be or not to be";fieldsep=" "; for(i=1; i<=NR;
i++)printf("$i\n")}'
```

(2)

```
$ awk 'BEGIN{sentence="To be or not to be"; fieldsep=" "; n=split(sentence, array,
fieldsep); for(i=1; i<=n; i++){printf("%s\n", array[i]);}}'
```

(3)

```
$ awk 'BEGIN{echo "To be or not to be"; fieldsep=" "; for(i=1; i<=NR;
i++)printf("$i\n")}'
```

(4)

```
$ awk 'BEGIN{echo "To be or not to be"; fieldsep=" "; n=split(sentence, array,
fieldsep); for(i=1; i<=n; i++){printf("%s\n", array[i]);}}'
```

Answer

(2)

Problem 9

Match the following expr operators to their use.

Operator	Use
1. index str1 str2	A. Returns the substring y characters in length starting at position x.
2. substr str x y	B. Returns the pattern match if 'str1' matches the pattern in 'str2'
3. match str1 str2	C. Returns the substring 'x' characters in length starting at position 'y'.
	D. Returns the pattern match if 'str2' matches the pattern in 'str1'
	E. Returns the starting position of substring 'str2' if found in the string 'str1'; else return 0
	F. Returns position in 'str1' if any of the characters given in 'str2' is found; else return 0

(1) 1-E, 2-D, 3-C

(2) 1-E, 2-B, 3-C

(3) 1-F, 2-B, 3-A

(4) 1-F, 2-B, 3-A

Answer

(4)

Problem 10

Which of the following scripts will give the output as below.[MSQ]

```
2,10
4,20
6,30
8,40
10,50
```

(1)

```
for (( i=2,j=10; i<11 && j<60; i+=2,j+=10)); do
  echo $i,$j
done
```

(2)

```
for (( i=2,j=10; i<11; i+=2,j+=10)); do
  echo $i,$j
done
```

(3)

```
for (( i=2; i<11; i+=2)); do
  for (( j=10; j<51; j+=10)); do
    echo $i,$j
  done
done
```

(4)

```
for (( i=2,j=10; i<11; i+=2)); do
  echo $i,$j
  j+=10
done
```

(5)

```
for (( i=2,j=10; i<11 || j<60;)); do
  echo $i,$j
  ((j+=10))
  ((i+=2))
done
```

Answer

(2) and (5)

Week 6 - Practice Assignmet

Problem 1

What will be the format of the output of the following command? [MCQ]

```
ls -l | awk '{print $1, $NF}'
```

(1) File Permission string, User

(2) User, File Name

(3) File Permission, File Name

(4) File Permission, File Type

Answer

(3) File Permission, File Name

Problem 2

Which of the following commands will print the file, appending the line number to the starting of each line, irrespective of the data in the file? **[MSQ]**
(1)

```
awk 'END {print NR,$0}' employee_details.txt
```

(2)

```
awk 'BEGIN{FS=","}{print NR,$1}' employee_details.txt
```

(3)

```
$ awk '{print NR,$0}' employee_details.txt
```

Answer

(3)

Problem 3

What is the output of the command given below? **(MCQ)**

```
$ awk 'BEGIN{print index("Ubuntu", "un"); print index("System Commands", "abc")}'
```

(1)

```
3
Error
```

(2)

```
4
Error
```

(3)

```
3
0
```

(4)

```
4
0
```

Answer

(3)

Problem 4

The built in variable `NF` used in awk scripts is used to.

(1) Display the line number.

(2) Display the first field in a line.

(3) Display the last field in a line.

(4) Display the number of fields in a line.

Answer

(4)

Problem 5

What is the output of the following command? **(MCQ)**

```
$ echo "927.8 -8.314 -0.87" | awk 'BEGIN{FIELDWIDTHS="3 4 3"}{print $1,$2,$3}'
```

(1) `927 -8.3 -0.`
(2) `927 .8 - 8.3`
(3) `927 -8.314 -0.87`
(4) `927.8 -8.314 -0.87`

Answer

(4)

##