Answers Type : Equal

Text Areas : PlainText

Possible Answers :

0

Question Number : 245 Question Id : 640653770671 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Short Answer Question

At a threshold of 0.4, how many “True Negatives” is the AI engine predicting?

(Note: Enter the answer rounded to two decimal places. For example, if the answer is “1.234”, then enter it as “1.23”)

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

1

System Commands

Section Id :	64065353271
Section Number :	15
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	15
Number of Questions to be attempted :	15
Section Marks :	100
Display Number Panel :	Yes

Section Negative Marks :	0
Group All Questions :	No
Enable Mark as Answered Mark for Review and Clear Response :	Yes
Maximum Instruction Time :	0
Sub-Section Number :	1
Sub-Section Id :	640653112648
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 246 Question Id : 640653770672 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 0

Question Label : Multiple Choice Question

THIS IS QUESTION PAPER FOR THE SUBJECT "DIPLOMA LEVEL : SYSTEM COMMANDS (COMPUTER BASED EXAM)"

ARE YOU SURE YOU HAVE TO WRITE EXAM FOR THIS SUBJECT?
CROSS CHECK YOUR HALL TICKET TO CONFIRM THE SUBJECTS TO BE WRITTEN.

(IF IT IS NOT THE CORRECT SUBJECT, PLS CHECK THE SECTION AT THE TOP FOR THE SUBJECTS REGISTERED BY YOU)

Options :

6406532577862. ✓ YES

6406532577863. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	640653112649
Question Shuffling Allowed :	Yes
Is Section Default? :	null

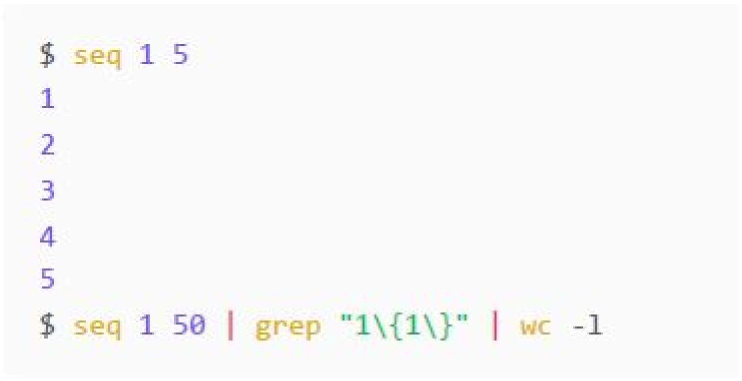
Question Number : 247 Question Id : 640653770673 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6

Question Label : Short Answer Question

What will be output from the given command?



```
$ seq 1 5
1
2
3
4
5
$ seq 1 50 | grep "1\{1\}" | wc -l
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

14

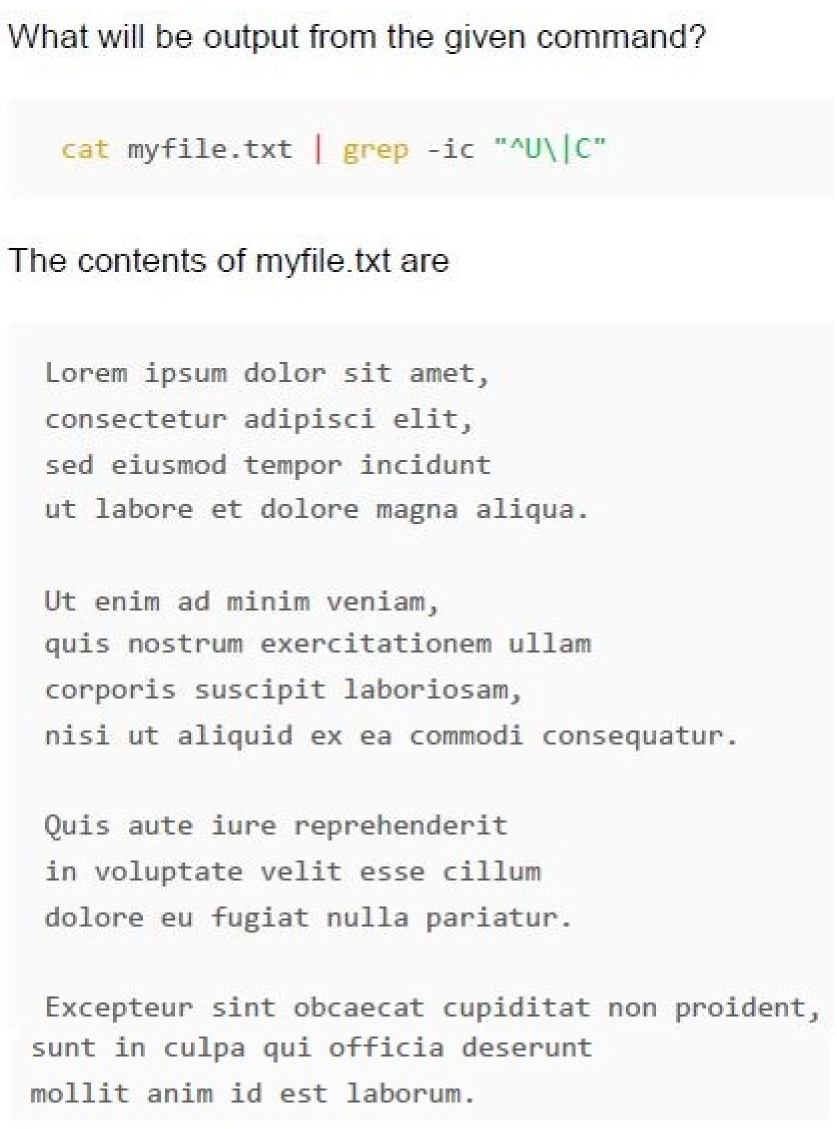
Sub-Section Number :	3
Sub-Section Id :	640653112650
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 248 Question Id : 640653770674 Question Type : SA Calculator : None

Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 8

Question Label : Short Answer Question


What will be output from the given command?

```
cat myfile.txt | grep -ic "^U\|C"
```

The contents of myfile.txt are

```
Lorem ipsum dolor sit amet,
consectetur adipisci elit,
sed eiusmod tempor incidunt
ut labore et dolore magna aliqua.

Ut enim ad minim veniam,
quis nostrum exercitationem ullam
corporis suscipit laboriosam,
nisi ut aliquid ex ea commodi consequatur.

Quis aute iure reprehenderit
in voluptate velit esse cillum
dolore eu fugiat nulla pariatur.

Excepteur sint obcaecat cupiditat non proident,
sunt in culpa qui officia deserunt
mollit anim id est laborum.
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

10

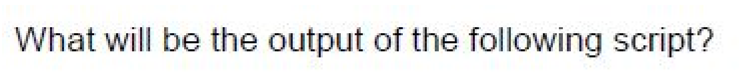
Sub-Section Number :	4
Sub-Section Id :	640653112651
Question Shuffling Allowed :	Yes
Is Section Default? :	null

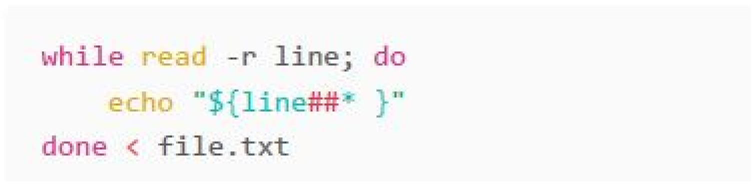
Question Number : 249 Question Id : 640653770675 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 6

Question Label : Multiple Choice Question


What will be the output of the following script?


```
while read -r line; do
    echo "${line##* }"
done < file.txt
```

Options :

6406532577866. ✖ It will print the first word present in the file

6406532577867. ✖ It will print the first word of each line present in the file

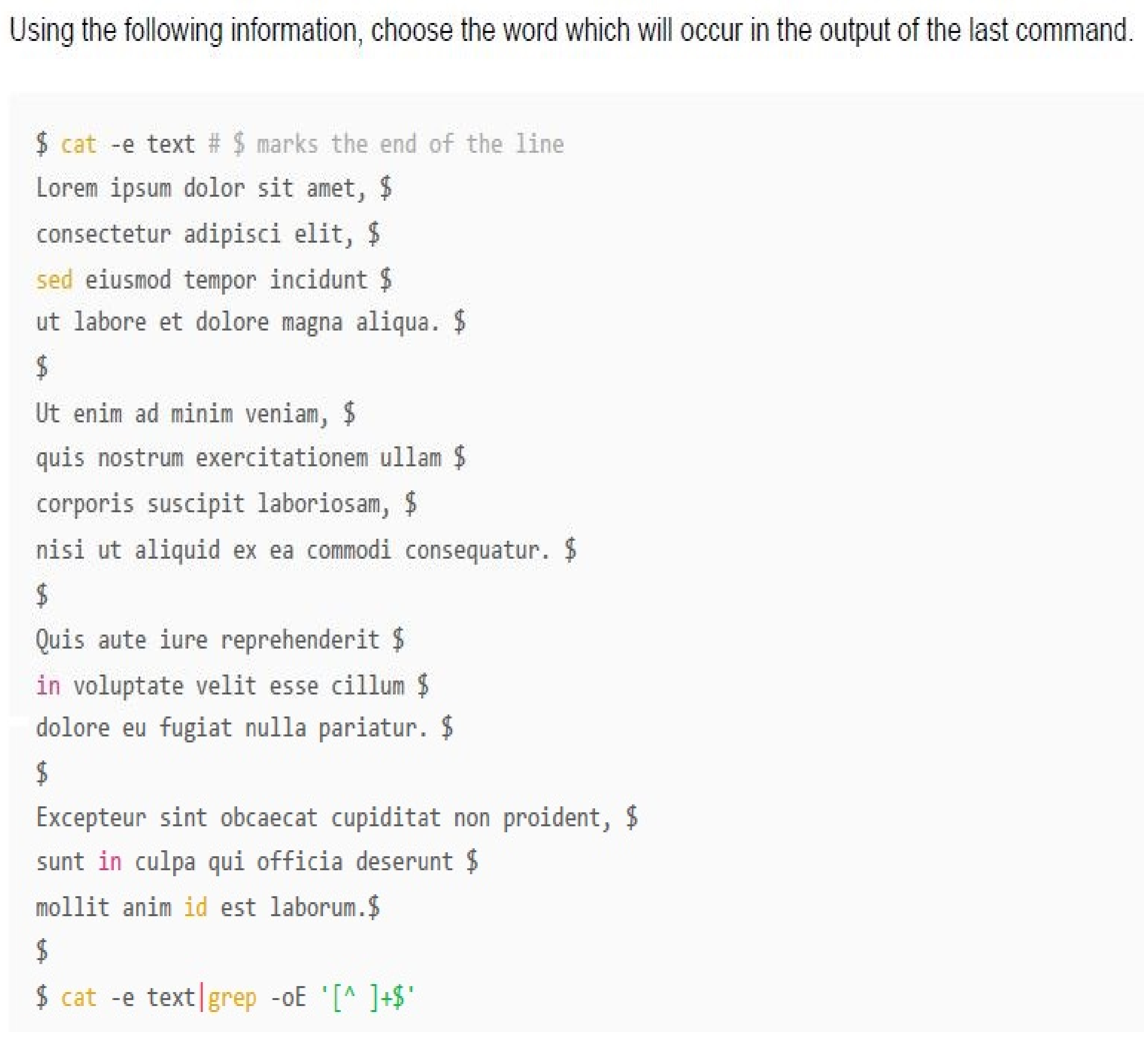
6406532577868. ✖ It will print the last word present in the file

6406532577869. ✓ It will print the last word of each line present in the file

Question Number : 250 Question Id : 640653770676 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6

Question Label : Multiple Choice Question



Using the following information, choose the word which will occur in the output of the last command.

```
$ cat -e text # $ marks the end of the line
Lorem ipsum dolor sit amet, $
consectetur adipisci elit, $
sed eiusmod tempor incidunt $
ut labore et dolore magna aliqua. $
$
Ut enim ad minim veniam, $
quis nostrum exercitationem ullam $
corporis suscipit laboriosam, $
nisi ut aliquid ex ea commodi consequatur. $
$
Quis aute iure reprehenderit $
in voluptate velit esse cillum $
dolore eu fugiat nulla pariatur. $
$
Excepteur sint obcaecat cupiditat non proident, $
sunt in culpa qui officia deserunt $
mollit anim id est laborum.$
$
$ cat -e text|grep -oE '[^ ]+$'
```

Options :

6406532577870. ✖ `aliqua. $`

6406532577871. ✖ `consequatur. $`

6406532577872. ✖ `pariatur. $`

6406532577873. ✓ `laborum.$`

6406532577874. ✖ all of these

Sub-Section Number : 5

Sub-Section Id : 640653112652

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 251 Question Id : 640653770677 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 8 Max. Selectable Options : 0

Question Label : Multiple Select Question

The current working directory has files "1.txt", "2.txt", ... "100.txt". Select the shell script(s) that will rename all the files to "1.md", "2.md", ... "100.md" respectively.

Hint:

xargs

`xargs` is a command that reads items from the standard input, delimited by blanks or newlines, and executes the command (default is /bin/echo) one or more times with any initial-arguments followed by items read from standard input.

- `-I` {} is used to replace the string {} with the input from standard input.
- **Example**: `ls | xargs cat` will print the content of all the files present in the current directory.

find

`find` command is used to search and locate the list of files and directories based on conditions you specify for files that match the arguments.

- `-exec` command is used to perform an action on the found files.
- `\;` is used to end the command.
- `-name` is used to search the files based on the name.

Options :

6406532577875. ✓ `for i in {1..100}; do mv $i.txt $i.md; done`

6406532577876. ✓ `for i in {1..100}; do cp $i.txt $i.md; done`

6406532577877. ✓ `ls *.txt | xargs -I {} mv {} {}.md`

6406532577878. ✓ `find . -name "*.txt" -exec mv {} {}.md \;`

Sub-Section Number : 6

Sub-Section Id : 640653112653

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 252 Question Id : 640653770678 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 7

Question Label : Multiple Choice Question

The current working directory has files "1.txt", "2.txt", ... "100.txt". Print the file's content in the order "100.txt", "99.txt", ... , "2.txt", "1.txt" respectively.

Hint:

ls

`ls` command is used to list the files and directories in the current directory.

- `-r` is used to reverse the order while sorting (using string comparison).

sort

`sort` command is used to sort the lines of a text file(s).

- `-r` is used to reverse the order while sorting.
- `-n` is used to compare according to string numerical value.

xargs

`xargs` is a command that reads items from the standard input, delimited by blanks or newlines, and executes the command (default is /bin/echo) one or more times with any initial-arguments followed by items read from standard input.

- **Example**: `ls | xargs cat` will print the content of all files present in the current directory.

Options :

6406532577879. ✖ `cat $(ls -r *.txt)`

6406532577880. ✖ `ls | sort -r | xargs cat`

6406532577881. ✗
```
cat $(ls *.txt)
```

6406532577882. ✓
```
ls | sort -rn | xargs cat
```

Sub-Section Number :	7
Sub-Section Id :	640653112654
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 253 Question Id : 640653770679 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 8 Max. Selectable Options : 0

Question Label : Multiple Select Question

The current working directory has files "1.txt", "2.txt", ... "100.txt". Select the script(s) to print the name of all the files such that each such file contains the name of the file as the sole content.

Example:

If `file1.txt` is printed by the script then the content of file `1.txt` is

```
1.txt
```

Options :

6406532577883. ✓
```
for i in {1..100}; do
    if [ "$(cat $i.txt)" = "$i.txt" ]; then
        echo $i.txt
    fi
done
```

6406532577884. ✓
```
for i in *.txt; do
    if [ "$(cat $i.txt)" = "$i.txt" ]; then
        echo "$(cat $i.txt)"
    fi
done
```

6406532577885. ✓
```
ls *.txt | while read i; do
    [[ "$(cat $i)" == "$i" ]] && echo $i
done
```

6406532577886. ✖
```
ls *.txt | xargs -I {} sh -c '[[ "$(cat {})" != "{}" ]] && echo {}'
```

Sub-Section Number : 8

Sub-Section Id : 640653112655

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 254 Question Id : 640653770680 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 10 Max. Selectable Options : 0

Question Label : Multiple Select Question

The current working directory has files "1.txt", "2.txt", ... "100.txt". Select the command(s) that will remove the files "70.txt", "71.txt", ... "99.txt".

Options :

6406532577887. ✓
```
rm {70..99}.txt
```

6406532577888. ✓
```
rm 7{0..9}.txt
```

6406532577889. ✓
```
rm 7?.txt; rm 8?.txt; rm 9?.txt
```

6406532577890. ✗ `rm [780]*.txt`

6406532577891. ✗ `rm 7*.txt; rm 8*.txt; rm 9*.txt`

6406532577892. ✓ `rm $(ls | grep -E "7[0-9].txt|8[0-9].txt|9[0-9].txt")`

6406532577893. ✓ `rm $(ls | grep -E "7[0-9]|8[0-9]|9[0-9].txt")`

6406532577894. ✓ `rm $(ls | grep -E "[789][0-9].txt")`

6406532577895. ✓ `rm $(ls | grep "7[0-9].txt\|8[0-9].txt\|9[0-9].txt")`

6406532577896. ✓ `for i in {70..99}; do rm $i.txt; done`

6406532577897. ✓ `ls | while read i; do [[ $i -ge 70 && $i -le 99 ]] && rm $i.txt; done`

6406532577898. ✓ `ls | while read i; do (( $i >= 70 && $i <= 99 )) && rm $i.txt; done`

Sub-Section Number :	9
Sub-Section Id :	640653112656
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 255 Question Id : 640653770681 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6 Max. Selectable Options : 0

Question Label : Multiple Select Question

Using the following information, which of the command(s) from the following options will produce the desired output shown?

```
$ df -m
Filesystem     1M-blocks    Used     Available Use%  Mounted on
/dev/sda1         102400    2048     100352    3%    /
/dev/sda2         204800    4096     200704    3%    /home
tmpfs             4096      0        4096      0%    /tmp
/dev/cdrom        1024      512      512       50%   /media/cdrom
/dev/sdb1         307200    102400   204800    34%   /mnt/data
/dev/sdc1         512000    102400   409600    20%   /mnt/backup
```

Desired Output

```
211456
```

Options :

6406532577899. ✓
```
df -m|awk '{p+=$3}; END {print p}'
```

6406532577900. ✖
```
df -m|awk '{p=+$3}; END {print p}'
```

6406532577901. ✓
```
df -m | awk 'NR>1 && $1 !~ /^(tmpfs|cdrom)$/ {sum += $3} END {print sum}'
```

6406532577902. ✖
```
df -m | awk 'NR>1 && $1 !~ /^(tmpfs|cdrom)$/ {sum =+ $3} END {print sum}'
```

Sub-Section Number : 10

Sub-Section Id : 640653112657

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 256 Question Id : 640653770682 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction

Time : 0

Correct Marks : 8 Max. Selectable Options : 0

Question Label : Multiple Select Question

Select the command(s) that list all regular users in the system. Assume that regular users have their UID greater than 999 and their default shell is bash (/usr/bin/bash).
Note: `-E` enables the Extended Regular Expression (ERE) in sed.

The file /etc/passwd contains the user information. The format of the file is given below

```
username:x:UID:GID:Description:Home Directory:Full Path to Shell
```

Hint
Usage of the `tr` command

```
$ echo 'a,b,c,d' | tr ',' '\n'
a
b
c
d
```

Options :

6406532577903. ✗ `sed -nE '/.+:.:[[:digit:]]{4,}:.*bash/ p' /etc/passwd`

6406532577904. ✓ `sed -nE '/.+:.:[[:digit:]]{4,}:.*bash/ p' /etc/passwd | cut -d: -f1`

6406532577905. ✗ `sed -nE '/.+:.:[[:digit:]]{3}:.*bash/ p' /etc/passwd`

6406532577906. ✗ `sed -nE '/.+:.:[[:digit:]]{3}:.*bash/ p' /etc/passwd | cut -d: -f1`

6406532577907. ✗ `awk '$3 > 999 && $7 ~ /.*bash/ {print $1}' /etc/passwd`

6406532577908. ✓ `awk -F ":" '$3 > 999 && $7 ~ /.*bash/ {print $1}' /etc/passwd`

6406532577909. ✓ `cat /etc/passwd|tr ':' '\t'|awk '$3 > 999 && $7 ~ /.*bash/ {print $1}'`

6406532577910. ✓ `awk 'BEGIN{FS=":"} $3 > 999 && $7 ~ /.*bash/ {print $1}' /etc/passwd`

Sub-Section Number :	11
Sub-Section Id :	640653112658
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 257 Question Id : 640653770683 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6

Question Label : Multiple Choice Question

Select the correct statement(s) based on the below script. Assume that `file1` is not empty.

Hint:

The `tee` command not only takes the stdin and prints it to the terminal but also writes to the file given as an argument.

```
while read line; do
    echo $line
done < file1 > file2 | tee file3
```

Options :

6406532577911. ✖ `file2` will be empty at the end of the execution

6406532577912. ✓ `file3` will be empty at the end of the execution

6406532577913. ✖ The contents of `file1` will be displayed in the terminal

6406532577914. ✖ `file3` will contain the contents of `file1`

Question Number : 258 Question Id : 640653770684 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 6

Question Label : Multiple Choice Question

The tab-separated data file `pdata.txt` provided for cleanup showed every fourth and fifth line is a part of one line (first-merge is the first and the second line) (see the following examples in the hint). Choose the correct commands from the following options, which can merge every fourth and fifth (1-2, 5-6, 9-10, 13-14, etc...) line into one line to clean up the data file.

Hint:

Use the following information.

```
$ cat pdata.txt
Line1
Line2
Line3
Line4
Line5
Line6
Line7
Line8
Line9
Line10
Line11
Line12
Line13
Line14
$ sed 'N;s/\n/ /' pdata.txt
Line1   Line2
Line3   Line4
Line5   Line6
Line7   Line8
Line9   Line10
Line11  Line12
Line13  Line14
$ sed 'N;N;s/\n/ /' pdata.txt
Line1   Line2
Line3
Line4   Line5
Line6
Line7   Line8
Line9
Line10  Line11
Line12
Line13  Line14
Line15
```

Options :

6406532577915. ✖ `sed 'N;N;N;s/\n/ /' pdata.txt`

6406532577916. ✖ `sed -i 'N;N;N;N;s/\n/ /' pdata.txt`

6406532577917. ✓ `sed -i 'N;N;N;s/\n/\t/' pdata.txt`

6406532577918. ✖ `sed -i 'N;N;N;N;s/\n/\t/' pdata.txt`

Sub-Section Number : 12

Sub-Section Id : 640653112659

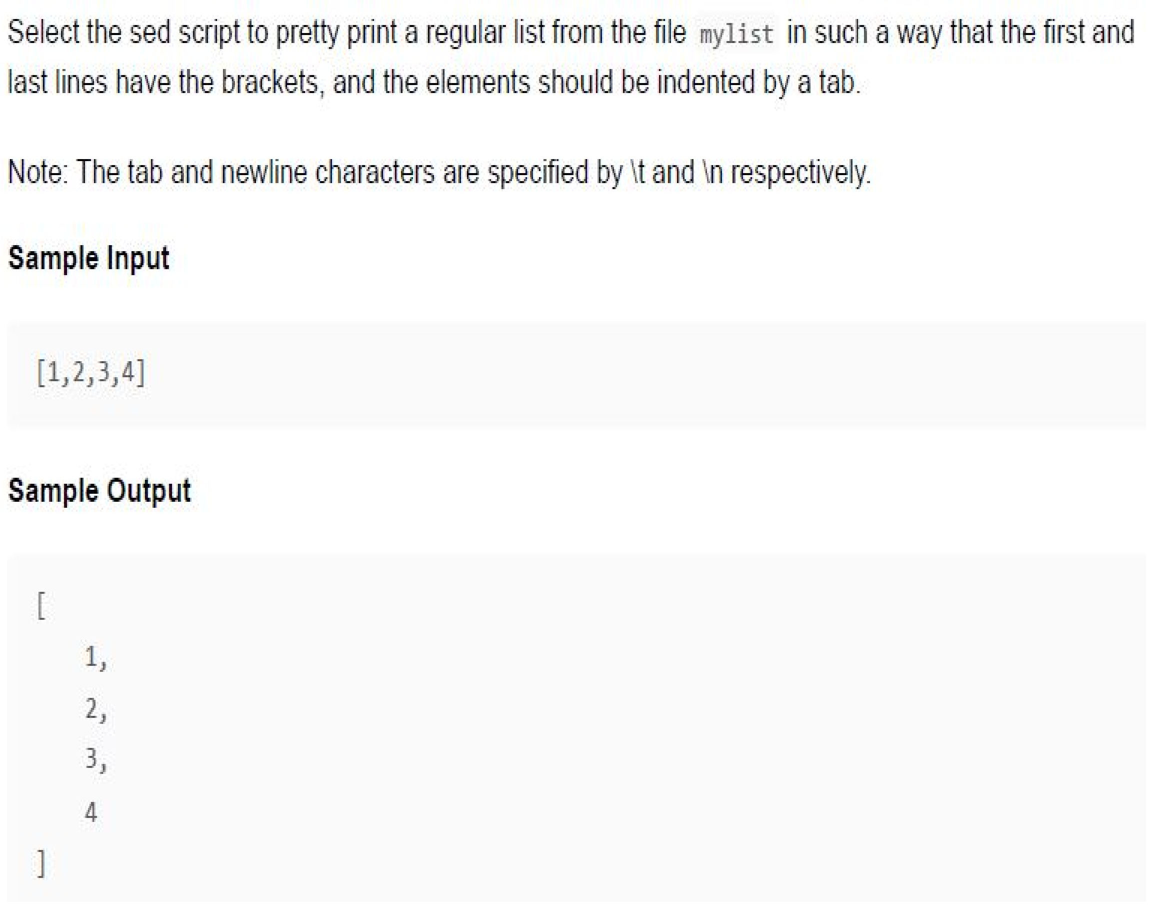
Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 259 Question Id : 640653770685 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 7

Question Label : Multiple Choice Question



Select the sed script to pretty print a regular list from the file `mylist` in such a way that the first and last lines have the brackets, and the elements should be indented by a tab.

Note: The tab and newline characters are specified by \t and \n respectively.

Sample Input

```
[1,2,3,4]
```

Sample Output

```
[
    1,
    2,
    3,
    4
]
```

Options :

6406532577919. ✓

```
sed 's/\[/\[\n\t/g' mylist |
sed 's/\]/\n]/' |
sed '/^[[:blank:]]\{1,\}/ s/,/,\n\t/g'
```

6406532577920. ✖
```
sed 's/\[/\[\n\t/g' mylist |
sed 's/\]/\n]/' |
sed '/^[[:blank:]]\{1,\}/ s/,/,\n\t/'
```

6406532577921. ✖
```
sed 's/\[/\[\t\n/g' mylist |
sed 's/\]/\n]/' |
sed '/^[[:blank:]]\{1,\}/ s/,/,\n\t/g'
```

6406532577922. ✖
```
sed 's/\[/\[\t\n/g' mylist |
sed 's/\]/\n]/' |
sed '/^[[:blank:]]\{1,\}/ s/,/,\n\t/'
```

Sub-Section Number :	13
Sub-Section Id :	640653112660
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 260 Question Id : 640653770686 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 8

Question Label : Multiple Choice Question

```
BEGIN {
    FS=","
}
{
    sum = 0
    for (i=1; i<=NF; i++) {
        if ($i ~ /^[-+]?[[:digit:]]+\.?[[:digit:]]*$/) {
            sum += $i
        }
        else {
            print "Invalid data"
            exit 1
        }
    }
    print sum
}
```

Select the output to the above AWK script for the file given below.

```
1,2,3
1.1,2.1,3.1
-1.1,2.1,3.1
+1.1,2.1,3.1
a,b,2
.1,89,1
```

Options :

6406532577923. ✖

```
Invalid data
```

6406532577924. ✔

```
6
6.3
4.1
6.3
Invalid data
```

6406532577925. ✖

```
6
6.3
4.1
6.3
2
Invalid data
```

6406532577926. ✖

```
6
6.3
4.1
6.3
2
1
```