

6406532330498. ✖ [0 0 0 1 0]

6406532330499. ✖ [0 1 0 0 0]

6406532330500. ✔ [0 1 0 1 0]

Question Number : 212 Question Id : 640653697794 Question Type : SA Calculator : None

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

Correct Marks : 3

Question Label : Short Answer Question

What will be the output of following code?

```
print(clf.score(X_test[30:35], y_test[30:35]))
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Range

Text Areas : PlainText

Possible Answers :

0.78 to 0.82

System Commands

Section Id :	64065349268
Section Number :	9
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
Section Marks :	105
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 :	640653103312
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 213 Question Id : 640653697795 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 :

6406532330502.  YES

6406532330503.  NO

Sub-Section Number :	2
Sub-Section Id :	640653103313
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 214 Question Id : 640653697796 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 following command is run in the terminal; which statement(s) are true from the option?

```
grep -xF -f <(sort file1.txt) file2.txt | sort | uniq
```

Hint

- x matches whole lines.
- F treats the patterns as fixed strings.
- f reads the patterns from the specified file (output of sort file1.txt).

```
$ seq 5
1
2
3
4
5
$ grep '[23]' <(seq 5) # the command output stored in temporary file, and that
file path is substituted in place of<(seq 5)
2
3
```

Options :

6406532330504. ✓ The `sort file1.txt` will be executed first and provide sorted input to the `grep`.

6406532330505. ✓ `grep` will try to find a match to each line from the process substitution output with `file2.txt`.

6406532330506. ✗ The command will fail due to the wrong implementation of process substitution.

6406532330507. ✓ The command is correct and will produce the common lines in `file1.txt` and `file2.txt`.

Question Number : 215 Question Id : 640653697801 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 retrieves the first ten lines of all the files that only end with `.md` in the current working directory and its subfolders.

Note: All the directories and files do not have space in their names

Hint:

- `-type f` option searches only for the files.
- `-name x*` option searches for the filename with the pattern with wildcard characters (not regex)

```
$ xargs --help
Usage: xargs [OPTION]... COMMAND [INITIAL-ARGS]...
Run COMMAND with arguments INITIAL-ARGS and more arguments read from input.

Mandatory and optional arguments to long options are also
mandatory or optional for the corresponding short option.

-0, --null                items are separated by a null, not whitespace;
                           disables quote and backslash processing and
                           logical EOF processing

-a, --arg-file=FILE       read arguments from FILE, not standard input

-d, --delimiter=CHARACTER items in input stream are separated by CHARACTER,
                           not by whitespace; disables quote and backslash
                           processing and logical EOF processing

-E END                    set logical EOF string; if END occurs as a line
                           of input, the rest of the input is ignored
                           (ignored if -0 or -d was specified)

-e, --eof[=END]           equivalent to -E END if END is specified;
                           otherwise, there is no end-of-file string

-I R                      same as --replace=R

-i, --replace[=R]         replace R in INITIAL-ARGS with names read
                           from standard input, split at newlines;
                           if R is unspecified, assume {}

-L, --max-lines=MAX-LINES use at most MAX-LINES non-blank input lines per
                           command line

...

$ head --help
Usage: head [OPTION]... [FILE]...
Print the first 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.

With no FILE, or when FILE is -, read standard input.
...
```

Options :

6406532330531. ✓ `find . -type f -name '*.md' | xargs -L 1 head`

6406532330532. ✓ `head $(find . -type f -name '*.md')`

6406532330533.

✖ `find . -type f -name '*.md*' | xargs -L 1 head`

6406532330534. ✖ `find . type f | grep md | head`

Sub-Section Number : 3
Sub-Section Id : 640653103314
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 216 Question Id : 640653697797 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 following command is executed in the terminal. What would be the output of the command?

```
cat date.txt | sort | tail -1
```

Sample Input

```
$ cat date.txt
Jun 24,2009
April 4,2010
April 14,2009
April 24,2010
May 4,2010
May 14,2010
May 24,2010
Jun 4,2010
Jun 14,2009
Jun 24,2010
```

Hint

By default sort will use string-based sort and not numerical sort.

Options :

6406532330508. ✖ Jun 24,2009

6406532330509. ✖ April 4,2010

6406532330510. ✖ April 14,2009

6406532330511. ✖ April 24,2010

6406532330512. ✔ May 4,2010

6406532330513. ✖ May 14,2010

6406532330514. ✖ May 24,2010

6406532330515. ✖ Jun 4,2010

6406532330516. ✖ Jun 14,2009

6406532330517. ✖ Jun 24,2010

Question Number : 217 Question Id : 640653697808 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

Text

```
 Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Donec a diam lectus.
 Sed sit amet ipsum mauris.
 Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit.
 Donec et mollis dolor.
```

Choose the output after giving the key sequence `ddP` in the vi editor. The cursor is at the beginning of the first line.

The equivalent key sequence in emacs is `C-SPACE C-n C-w C-y`, where `C-SPACE` is the control key and the space bar pressed together.

Options :

6406532330556. ✖

Donec a diam lectus.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Sed sit amet ipsum mauris.
Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit.
Donec et mollis dolor.

6406532330557. ✔

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec a diam lectus.
Sed sit amet ipsum mauris.
Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit.
Donec et mollis dolor.

6406532330558. ✖

ipsum Lorem dolor sit amet, consectetur adipiscing elit.
Donec a diam lectus.
Sed sit amet ipsum mauris.
Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit.
Donec et mollis dolor.

6406532330559. ✖

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Donec a diam lectus.
Sed sit amet ipsum mauris.
Maecenas congue ligula ac quam viverra nec consectetur ante hendrerit.
Donec et mollis dolor.

Question Number : 218 Question Id : 640653697810 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

Following Shell Script is used to print the common lines between two files. Please choose the option that will correct the script.

```
#!/bin/bash
FILENAME1="$1"
FILENAME2="$2"
IFS='\\n'
for line1_file1 in $(cat "$FILENAME1")
do
    for line1_file2 in $(cat "$FILENAME2")
    do
        if [ "${line_file1}" == "${line_file2}" ]
        then
            echo "${line1_file1}"
        fi
    done
done
```

Options :

6406532330567. ✓ There should be a \$ before \\n in the IFS assignment

6406532330568. ✗ The call in for loop \$(cat \${FILENAME1}) is incorrect, and the while loop should have been used to read lines.

6406532330569. ✗ The if statement is incorrect as it has extra space between the [and \$ as well as } and] .

6406532330570. ✗ The code is correct and has no issues.

Sub-Section Number :

4

Sub-Section Id :

640653103315

Question Shuffling Allowed :

Yes

Is Section Default? :

null

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 **regular expression** (ERE) that matches the date before 2022-01-01 and the amount is less than 10000.00. The date is in YYYY-MM-DD format, and the amount is in decimal format with two digits after the decimal point.

Note: The year is from 2000 to 2023 (both inclusive).

Sample Input

```
Company,Date,Amount
CompanyA,2020-04-29,31211.18
CompanyA,2021-09-20,2366.91
CompanyA,2021-10-23,3491.17
CompanyB,2021-04-04,14304.44
CompanyA,2021-11-01,21679.58
CompanyA,2022-09-06,16277.49
CompanyA,2021-12-14,29527.37
CompanyA,2023-03-01,1447.83
CompanyA,2020-08-22,30383.23
```

Sample Output

```
CompanyA,2021-09-20,2366.91
CompanyA,2021-10-23,3491.17
```

Options :

6406532330518. ✖ `.*,20[12][01]-[01][0-9]-[0-9]{2},[0-9]{4}\.[0-9]{2}`

6406532330519. ✖ `.*,20[012][01]-[01][0-9]-[0-9]{2},[0-9]{5}\.[0-9]{2}`

6406532330520. ✔ `.*,20[012][01]-[01][0-9]-[0-9]{2},[0-9]{4}\.[0-9]{2}`

6406532330521. ✖

.*,202[1-9]-[01][0-9]-[0-9]{2},[0-9]{5}\.[0-9]{2}

Question Number : 220 Question Id : 640653697799 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

Which commands can be used to select the package name and its version as shown in the desired output from the following log file?

Note: Your solution should satisfy the sample output given the sample input.

Sample log file

```
2023-12-09 status installed nginx-common:all 1.18.0-6ubuntu14.4
2023-12-09 configure libnginx-mod-http-xslt-filter:amd64 1.18.0-6ubuntu14.4
<none>
2023-12-09 status unpacked libnginx-mod-http-xslt-filter:amd64 1.18.0-6ubuntu14.4
2023-12-09 status half-configured libnginx-mod-http-xslt-filter:amd64 1.18.0-6ubuntu14.4
```

Desired output

```
nginx-common:all
libnginx-mod-http-xslt-filter:amd64
libnginx-mod-http-xslt-filter:amd64
libnginx-mod-http-xslt-filter:amd64
```

Hint

```
# Relevant section from man grep

-E, --extended-regexp
        Interpret PATTERNS as extended regular expressions (EREs, see
below).
-o, --only-matching
        Print only the matched (non-empty) parts of a matching line, with
each such part on a separate output line.
The Backslash Character and Special Expressions
        The symbols \< and \> respectively match the empty string at the beginning
and end of a word. The symbol \b matches the empty string
        at the edge of a word, and \B matches the empty string provided it's not
at the edge of a word. The symbol \w is a synonym for
        [_[:alnum:]] and \W is a synonym for [^[:alnum:]].

Repetition
        A regular expression may be followed by one of several repetition
operators:
        ?      The preceding item is optional and matched at most once.
        *      The preceding item will be matched zero or more times.
        +      The preceding item will be matched one or more times.
        {n}     The preceding item is matched exactly n times.
        {n,}    The preceding item is matched n or more times.
        {,m}    The preceding item is matched at most m times. This is a GNU
extension.
        {n,m}   The preceding item is matched at least n times but not more than m
times.
```

Options :

6406532330522. ✖ `grep -oE '([[:alnum:]]-)*:[[:alnum:]]*' dpkg.log`

6406532330523. ✖ `grep -oE '[:alnum:]]-]+:[[:alnum:]]+' dpkg.log`

6406532330524. ✖ `grep -oE '[:alnum:]-*:[[:alnum:]]*' dpkg.log`

6406532330525. ✔ `grep -oE '([[:alnum:]-]+:[[:alnum:]]+)' dpkg.log`

Question Number : 221 Question Id : 640653697800 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

Anish wanted to extract phone numbers from the following data. He knows that **grep -oE** can print the pattern selectively using Extended regular expression. Please help him choose the correct regular expression from the options to achieve that.

Sample Input

```
name,phone,email,region
Madan,1-158-662-4996,madan-raja@outlook.ca,GA
Persaud,1-877-704-5869,persaud@google.edu,Meghalaya
Srivas,1-516-922-8416,k-srivas@icloud.com,HR
Swami,761-1395,swami@google.couk,Haryana
Subram,981-5610,subram3142@yahoo.org,Karnataka
Nirmal,474-7526,nirmal@icloud.org,Madhya Pradesh
Sahni,151-8534,sahni@hotmail.ca,KA
Mahajan,1-548-689-8736,mahajan@icloud.couk,Bihar
Rana,1-528-385-7783,rana4716@yahoo.org,AN
```

Sample Output

```
1-158-662-4996
1-877-704-5869
1-516-922-8416
761-1395
981-5610
474-7526
151-8534
1-548-689-8736
1-528-385-7783
```

Options :

6406532330526. ✖ `[0-9]{1}\-[0-9]{3}\-[0-9]{3}\-[0-9]{4}|[0-9]{3}\-[0-9]{4}`

6406532330527. ✖ `[0-9]{1}-[0-9]{3}-[0-9]{3}-[0-9]{4}|[0-9]{3}-[0-9]{4}`

6406532330528. ✖ `\b[0-9]-[0-9]{3}-[0-9]{3}-[0-9]{4}\b|\b[0-9]{3}-[0-9]{4}\b`

6406532330529.

✖ `\b[0-9].*[0-9]{3}-[0-9]{4}\b| [0-9]{3}-[0-9]{4}\b`

6406532330530. ✔ All of these

Question Number : 222 Question Id : 640653697802 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

What will be the output of the last command in the following sequence of commands?

```
$ seq 10
1
2
3
4
5
6
7
8
9
10
$ seq 10 | sed '1d;2d'
```

Options :

3
4
5
6
7
8
9
10

6406532330535. ✔

6406532330536. ✖

1
4
5
6
7
8
9
10

2
4
5
6
7
8
9
10

6406532330537. ✖

2
3
4
5
6
7
8
9

6406532330538. ✖

Question Number : 223 Question Id : 640653697804 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 sample input data is provided as follows. The actual file is very long. Vinisha wanted to see if she could filter data using two conditions using bash. She wanted to look at all the Sales and HR department entries with salaries higher than 50000. Which of the following options contains code that will achieve the filtering process?

Sample Input

```
$ cat salarydata.txt
```

ID	Name	Department	Salary
101	Alice	HR	55000
102	Bob	Sales	60000
103	Charlie	Marketing	58000
104	David	HR	52000
105	Emily	Sales	63000
106	Frank	Marketing	59000

Options :

6406532330543. ✓ `awk '($3 == "HR" || $3 == "Sales") && $4 > 50000' salarydata.txt`

6406532330544. ✗ `awk '$3 == "HR" || $3 == "Sales" && $4 > 50000' salarydata.txt`

6406532330545. ✗ `awk '($3 == "HR" && $3 == "Sales") || $4 > 50000' salarydata.txt`

6406532330546. ✗ `awk '$3 == "HR" && $3 == "Sales" || $4 > 50000' salarydata.txt`

Question Number : 224 Question Id : 640653697805 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 following awk command is executed on the terminal. Which of the following is true about the output of the command?

```
awk '/./ { count+=1 } END { print count }' file.txt
```

Options :

6406532330547. ✖ The output will be the count of all the lines with only one character present.

6406532330548. ✖ The output will be the count of all the empty lines present in the file.

6406532330549. ✔ The output will be the count of all the non-empty lines.

6406532330550. ✖ The output will be the count of the total number of lines present in the file.

Question Number : 225 Question Id : 640653697806 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 command that kills the process not owned by the user `root` and consumes the maximum CPU time. Assume you have the permission to kill any process.

Hint:

- The 10th column of the output of `ps aux` command (TIME) contains the total CPU time used by the process.
- The `ps aux` command can list all the processes owned by all the users.

```
$ ps aux | head -5 # get the first 5 lines
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  21416 13004 ?        Ss   11:26   0:00 /sbin/init
root         2  0.0  0.0      0     0 ?        S    11:26   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    11:26   0:00
[pool_workqueue_release]
priya     2336  0.0  0.0  11396  2744 tty1    S+   11:29   0:00 cat
$ ps aux | head -5 | sort -k2,2 -rn # sorted by second column in reverse order
priya     2336  0.0  0.0  11396  2744 tty1    S+   11:29   0:00 cat
root         3  0.0  0.0      0     0 ?        S    11:26   0:00
[pool_workqueue_release]
root         2  0.0  0.0      0     0 ?        S    11:26   0:00 [kthreadd]
root         1  0.0  0.1  21416 13004 ?        Ss   11:26   0:00 /sbin/init
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
```

Options :

```
kill $(
    ps aux |
    head -1 |
    sort -k10,10 -r |
    grep -v '^root\b' |
    sed 1d |
    awk '{print $2}'
)
```

6406532330551. ✖

```
kill $(
    ps aux |
    sed 1d |
    sort -k10,10 -rn |
    grep -v '^root\b' |
    head -1 |
    awk '{print $2}'
)
```

6406532330552. ✔

```
kill $(
    ps aux |
    sed 1d |
    sort -k10,10 -r |
    grep -v '^root\b' |
    head -1 |
    awk '{print $2}'
)
```

6406532330553. ✖

```
kill $(
    ps aux |
    grep -v '^root\b' |
    sort -k10,10 -rn |
    sed 1d |
    awk '{print $2}' |
    head -1
)
```

6406532330554. ✖

Question Number : 226 Question Id : 640653697809 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 following scripts are using various means to read a file into a script/loop. Please select the **incorrect statement** from the options.

```
#!/bin/bash

# Case 1
while read line
do
    echo $line
done < "data.txt"

# Case 2
cat data.txt|while read line
do
    echo $line
done

# Case 3
for line in $(cat data.txt)
do
    echo $line
done

# Case 4
while IFS=':' read -r name age country; do
    echo "Name: $name, Age: $age, Country: $country"
done < "data.txt"

# Case 5
while IFS= read -r line; do
    echo "Name: $line"
done < "data.txt"
```

Options :

- 6406532330560. ✖ Case 1 uses the input redirection method to read the file line-by-line.
- 6406532330561. ✖ Case 2 uses the pipe redirection to feed the cat output to the while loop.
- 6406532330562. ✖ The pipe redirection used in case 2 is inefficient compared to input redirection.
- 6406532330563. ✖ Case 3 uses command substitution with the for loop to read the file.
- 6406532330564. ✖ Case 4 is input redirection, which reads each entry in the data.txt file (assuming the file has three columns) into individual variables.
- 6406532330565.

✖ Case 5 reads the entire line of the data file and assigns it to the line variable.

6406532330566. ✔ None of the options are correct.

Sub-Section Number :	5
Sub-Section Id :	640653103316
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 227 Question Id : 640653697803 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 9

Question Label : Multiple Choice Question

Choose the sed script that converts each line of the input file to an element in the JSON object. The input file contains two words per line, the first word is the key and the second word is the value. The output should be a valid JSON object of key-value pairs.

Note:

- A valid JSON object is enclosed in curly braces.
- The key and value are separated by a colon.
- The key and value are enclosed in double quotes.
- Each key-value pair is separated by a comma.
- There is no comma after the last key-value pair.

Sample Input

```
fruit1,apple
fruit2,banana
fruit3,orange
```

Sample Output

```
{
"fruit1":"apple",
"fruit2":"banana",
"fruit3":"orange"
}
```

Options :

```
#!/usr/bin/sed -f
1 i {
$ s/^\(.*\) , \(.*\)$/"\1":"\2",/
$ a }
```

6406532330539. ✖

```
#!/usr/bin/sed -f
1 i {
$! s/^\(.*\) , \(.*\)$/"\1:\2",/
$ s/^\(.*\) , \(.*\)$/"\1:\2"/
$ a }
```

6406532330540. ✖

6406532330541.



```
#!/usr/bin/sed -f
1 i {
$! s/^\(.*\),\(.*\)$/"\1":"\2",/
$ s/^\(.*\),\(.*\)$/"\1":"\2"/
$ a }
```

```
#!/usr/bin/sed -f
1 a {
$ s/^\(.*\),\(.*\)$/"\1":"\2",/
$ i }
```

6406532330542. ✖

Sub-Section Number :	6
Sub-Section Id :	640653103317
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 228 Question Id : 640653697807 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

The files file1.txt and file2.txt are present in the current working directory with the file1.txt composed of four lines. How many lines does file2.txt have?

Note: The line 4c4,5 is not relevant to the solution.

```
$ diff file1.txt file2.txt
4c4,5
< mnop
---
> mnpo
> qrst
```

Response Type : Numeric
Evaluation Required For SA : Yes
Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

5

TDS

Section Id :	64065349269
Section Number :	10
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	21
Number of Questions to be attempted :	21
Section Marks :	20
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 :	640653103318
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 229 Question Id : 640653697811 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