**Question Number : 236 Question Id : 640653563947 Question Type : SA Calculator : None**

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

**Correct Marks : 2**

Question Label : Short Answer Question

Scrap Percentage after broaching process_____%

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Range

**Text Areas :** PlainText

**Possible Answers :**

3.15 to 3.25

**Question Number : 237 Question Id : 640653563948 Question Type : SA Calculator : None**

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

**Correct Marks : 2**

Question Label : Short Answer Question

Loss due to scrap after broaching_____

**Response Type :** Numeric

**Evaluation Required For SA :** Yes

**Show Word Count :** Yes

**Answers Type :** Range

**Text Areas :** PlainText

**Possible Answers :**

5050 to 5300

# System Commands

**Section Id :** 64065338326

| | |
|---|---|
| **Section Number :** | 9 |
| **Section type :** | Online |
| **Mandatory or Optional :** | Mandatory |
| **Number of Questions :** | 14 |
| **Number of Questions to be attempted :** | 14 |
| **Section Marks :** | 100 |
| **Display Number Panel :** | Yes |
| **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 :** | 64065380397 |
| **Question Shuffling Allowed :** | No |
| **Is Section Default? :** | null |

**Question Number : 238 Question Id : 640653563953 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 :**

6406531885193. ✔ YES

6406531885194. ✖ NO

**Sub-Section Number :** 2

**Sub-Section Id :** 64065380398

**Question Shuffling Allowed :** Yes

**Is Section Default? :** null

**Question Number : 239 Question Id : 640653563954 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

```
$ help echo
echo: echo [-neE] [arg ...]
    Write arguments to the standard output.

    Display the ARGs, separated by a single space character and followed
    by a newline, on the standard output.

    Options:
      -n    do not append a newline
      -e    enable interpretation of the following backslash escapes
      -E    explicitly suppress interpretation of backslash escapes

    `echo` interprets the following backslash-escaped characters:
      ......
      \n    new line
      ......
      \t    horizontal tab
      ......
```

What will be the output of the following script?

```
echo -ne "a\tb"
echo -ne " \tc"
echo -ne " \td"
echo -ne " \te\tf"
```

**Options :**

6406531885195. ✖

```
a	b
	c
	d
	e	f
```

6406531885196. ✔

```
a	b	c	d	e	f
```

6406531885197. ✖

```
a	b
c
d
e	f
```

6406531885198. ✖

```
a\tb
 \tc
 \td
 \te\tf
```

**Question Number : 240 Question Id : 640653563955 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 entry is made to a crontab. When is the script `/home/garima/premodel.sh` scheduled to get executed.

```
5 0 * * 1 /home/garima/premodel.sh
```

Hint: Below is the description of the sequence in the cron job command. It tells at what date/time periodically the job needs to be executed.

```
*   *   *   *   *   <Command(s) with argument>
|   |   |   |   |              |
|   |   |   |   |      Command or Script to Execute
|   |   |   |   |
|   |   |   |   |
|   |   |   |   |
|   |   |   | Day of the Week(0-6)
|   |   |   |
|   |   | Month of the Year(1-12)
|   |   |
|   | Day of the Month(1-31)
|   |
| Hour(0-23)
|
Min(0-59)
```

**Options :**

6406531885199. ✔ Every Monday at 00:05

6406531885200. ✖ Every Monday at 05:00

6406531885201. ✖ Everyday at 08:00

6406531885202. ✖ Everyday at 08:05 in May

**Question Number : 241 Question Id : 640653563969 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

file.csv

```
Sahni,151-8534,sahni@hotmail.ca,KA
Mahajan,1-548-689-8736,mahajan@icloud.co.uk,Bihar
Rana,1-528-385-7783,rana4716@yahoo.org,AN
```

Given a CSV file `file.csv` with contents as shown above. Select AWK command(s) which can produce the output as shown below.

Expected Output

```
Sahni
151-8534
Mahajan
1-548-689-8736
Rana
1-528-385-7783
```

Hint:

```
$ man awk | cat
...
FS   The input field separator,  a  space  by  default.
     See Fields, above.
...
OFS  The output field separator, a space by default.

ORS  The output record separator, by default a newline.
...
```

**Options :**

6406531885253. ✔ `awk -F, '{print $1"\n"$2}' file.csv`

6406531885254. ✖ `awk 'BEGIN{FS=",";OFS="\n"}{print $1 $2}' file.csv`

6406531885255. ✖ `awk 'BEGIN{FS=",";ORS="\n"}{print $1,$2}' file.csv`

6406531885256. ✖ `awk -F, '{print $1"\n",$2}' file.csv`

**Question Number : 242 Question Id : 640653563970 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

In a working directory number of text files are present with varying sizes (varying number of lines). For the following AWK script executed in the working directory, what is true for the output from the following options?

```
#!/usr/bin/awk -f

n < FNR {
    n = FNR
    nf = FILENAME
}
END {
    print nf
}
```

Hint:

```
$ man awk | cat
...
FILENAME The  name  of the current input file.  If no files
         are specified on the command line,  the  value  of
         FILENAME  is  “-”.  However, FILENAME is undefined
         inside the BEGIN rule (unless set by getline).

FNR      The input record number in the current input file.
...
```

**Options :**

6406531885257. ✔ The file name with the maximum number of lines among the files that are passed as arguments is printed

6406531885258. ✖ The file name that has the minimum number of lines among the files that are passed as arguments is printed

6406531885259. ✖ The file name that is passed as the first file argument is printed

6406531885260. ✖ The file name that is passed as the last file argument is printed

**Sub-Section Number :** 3

**Sub-Section Id :** 64065380399

**Question Shuffling Allowed :** Yes

**Is Section Default? :** null

**Question Number : 243 Question Id : 640653563960 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

```
while read -r line; do
    if [[ $line =~ ^[[:digit:]].*[[:digit:]]$ ]]; then
        echo $line
    fi
done < file1 > file2
diff file1 file2
```

Select a scenario from the following options such that upon execution of the above script, no lines will be printed to the terminal.

**Options :**

6406531885219. ✖ file1 and file2 have the same number of lines

6406531885220. ✖ file2 being the copy of file1

6406531885221. ✖ file2 containing all the lines that start and end with a number

6406531885222. ✔ file1 containing all the lines that start and end with a number

**Sub-Section Number :** 4

**Sub-Section Id :** 64065380400

**Question Shuffling Allowed :** Yes

**Is Section Default? :** null

**Question Number : 244 Question Id : 640653563968 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

What will be the output of the following command?

```
echo {a..c}{1..3} | tr ' ' '\n' | awk '
{
    count+=NF
    count2+=NR
    count3+=length+1
}
END {
    print count, count2, count3
}
'
```

Hint:

```
$ tr --help
Usage: tr [OPTION]... SET1 [SET2]
Translate, squeeze, and/or delete characters from standard input,
writing to standard output.
...

$ awk --help
Usage: awk [POSIX or GNU style options] -f progfile [--] file ...
Usage: awk [POSIX or GNU style options] [--] 'program' file ...
POSIX options:          GNU long options: (standard)
        -f progfile             --file=progfile
        -F fs                   --field-separator=fs
...

$ man awk | cat
...
length([s]) Return  the length of the string s, or
            the length of $0 if s is not supplied.
            As  a  non-standard extension, with an
            array argument, length()  returns  the
            number of elements in the array.
...
```

**Options :**

6406531885249. ✔ 9 45 27

6406531885250. ✖ 9 9 18

6406531885251. ✖ 9 27 27

6406531885252. ✖ 9 27 18

**Sub-Section Number :** 5

**Sub-Section Id :** 64065380401

**Question Shuffling Allowed :** Yes

**Is Section Default? :** null

**Question Number : 245 Question Id : 640653563961 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 6 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following command would print lines with Alice or Rabbit with total count of their occurrences at the end of the output from `alice.txt` file.
Hint:

- The pipe character do not have special meaning in Basic Regular Expression Engine(BRE) unless escaped with backslash
- -E option in grep enables Extended Regular Expression Engine(ERE)
- -c option gives the count of the line that have matches

**Options :**

6406531885223. ✖ `grep 'Alice|Rabbit' alice.txt | grep -c 'Alice|Rabbit'`

6406531885224. ✖ `grep -E 'Alice|Rabbit' alice.txt| grep -E -c 'Alice|Rabbit'`

6406531885225. ✔ `grep 'Alice\|Rabbit' alice.txt; grep -c 'Alice\|Rabbit'`

6406531885226. ✔ `grep -E 'Alice|Rabbit' alice.txt; grep -E -c 'Alice|Rabbit'`

**Question Number : 246 Question Id : 640653563966 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 6 Selectable Option : 0**

Question Label : Multiple Select Question

Given the file `test.csv` with content shown below. Select the sed command(s) that will filter lines with abbreviated regions.

```
$ cat test.csv
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
```

Expected output

```
Madan,1-158-662-4996,madan-raja@outlook.ca,GA
Srivas,1-516-922-8416,k-srivas@icloud.com,HR
Sahni,151-8534,sahni@hotmail.ca,KA
Rana,1-528-385-7783,rana4716@yahoo.org,AN
```

**Options :**

6406531885241. ✔ `sed -n '/[A-Z]\{1\}$/ p' test.csv`

6406531885242. ✔ `sed -n '/,[[:upper:]]\{2\}$/ p' test.csv`

6406531885243. ✖ `sed -n '/[[:alnum:]]\{2\}$/ p' test.csv`

6406531885244. ✔ `sed -n '/[[:upper:]]\{2\}$/ p' test.csv`

**Question Number : 247 Question Id : 640653563967 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 6 Selectable Option : 0**

Question Label : Multiple Select Question

Which of the following SED command(s) would print lines with Alice or Rabbit from `alice.txt` file.

Hint:

- The pipe character do not have special meaning in Basic Regular Expression Engine(BRE) unless escaped with a backslash
- -E option in grep enables Extended Regular Expression Engine(ERE)
- -n option prevent the default printing in sed

**Options :**

6406531885245. ✖ `sed -E 's/Alice\|Rabbit//' alice.txt`

6406531885246. ✖ `sed -E -n 's/Alice|Rabbit//' alice.txt`

6406531885247. ✔ `sed -n '/Alice\|Rabbit/ p' alice.txt`

6406531885248. ✔ `sed -En '/Alice|Rabbit/ p' alice.txt`

| | |
|---|---|
| **Sub-Section Number :** | 6 |
| **Sub-Section Id :** | 64065380402 |
| **Question Shuffling Allowed :** | Yes |
| **Is Section Default? :** | null |

**Question Number : 248 Question Id : 640653563959 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 8 Selectable Option : 0**

Question Label : Multiple Select Question

Contents of the current working directory are given below.

```
$ ls -R
.:
a  b

./a:
file0  file1  file2  file3  file4

./b:
file10  file3  file4  file5  file6
```

Select all the file(s) that will be present in the current working directory after the execution of the following script.

```
cd a
for i in *; do
    ls ../b | grep $i && mv $i ../b
done
```

**Options :**

6406531885211. ✔ file0

6406531885212. ✖ file1

6406531885213. ✔ file2

6406531885214. ✖ file3

6406531885215. ✖ file4

6406531885216. ✖ file5

6406531885217. ✖ file6

6406531885218. ✖ file10

**Question Number : 249 Question Id : 640653563962 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 8 Selectable Option : 0**

Question Label : Multiple Select Question

For the given regular expression (regex) identify the correct statement(s) from the following options. Note that the Extended Regular Expression (ERE) is used.

Hint:
The group number starts from the outer to inner and left to right respectively.
Note: a word, phrase, or sequence that reads the same backwards as forwards e.g. Noon, Anna

```
^((.)\2|(.).\3|(.)(.)\5\4)$
```

**Options :**

6406531885227. ✖ This regex will match with 1 character palindrome

6406531885228. ✔ This regex will match with 2 character palindrome

6406531885229. ✔ This regex will match with 3 character palindrome

6406531885230. ✔ This regex will match with 4 character palindrome

| | |
|---|---|
| **Sub-Section Number :** | 7 |
| **Sub-Section Id :** | 64065380403 |
| **Question Shuffling Allowed :** | No |
| **Is Section Default? :** | null |

**Question Id : 640653563956 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Question Numbers : (250 to 251)**

Question Label : Comprehension

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

$ seq 5
1
2
3
4
5

$ seq 5 | xargs echo
1 2 3 4 5

$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic
characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing
them;
                               e.g., '--block-size=M'; see SIZE format
below
  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of
last
                               modification of file status information);
                               with -l: show ctime and sort by name;
                               otherwise: sort by ctime, newest first
  -C                         list entries by columns
      --color[=WHEN]         colorize the output; WHEN can be 'always'
```

```
(default
                                if omitted), 'auto', or 'never'; more info
below
  -d, --directory             list directories themselves, not their
contents
  -D, --dired                 generate output designed for Emacs' dired
mode
  -f                          do not sort, enable -aU, disable -ls --color
  -F, --classify              append indicator (one of */=>@|) to entries
      --file-type             likewise, except do not append '*'
      --format=WORD           across -x, commas -m, horizontal -x, long -
l,
                                single-column -1, verbose -l, vertical -C
      --full-time             like -l --time-style=full-iso
  -g                          like -l, but do not list owner
      --group-directories-first
                              group directories before files;
                                can be augmented with a --sort option, but
any
                                use of --sort=none (-U) disables grouping
  -G, --no-group              in a long listing, don't print group names
  -h, --human-readable        with -l and -s, print sizes like 1K 234M 2G
etc.
      --si                    likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                              follow symbolic links listed on the command
line
      --dereference-command-line-symlink-to-dir
                              follow each command line symbolic link
                                that points to a directory
      --hide=PATTERN          do not list implied entries matching shell
PATTERN
                                (overridden by -a or -A)
      --hyperlink[=WHEN]      hyperlink file names; WHEN can be 'always'
                                (default if omitted), 'auto', or 'never'
      --indicator-style=WORD  append indicator with style WORD to entry
names:
                                none (default), slash (-p),
                                file-type (--file-type), classify (-F)
  -i, --inode                 print the index number of each file
  -I, --ignore=PATTERN        do not list implied entries matching shell
PATTERN
  -k, --kibibytes             default to 1024-byte blocks for disk usage;
```

```
                               used only with -s and per directory totals
  -l                           use a long listing format
  -L, --dereference            when showing file information for a symbolic
                               link, show information for the file the
link
                               references rather than for the link itself
  -m                           fill width with a comma separated list of
entries
  -n, --numeric-uid-gid        like -l, but list numeric user and group IDs
  -N, --literal                print entry names without quoting
  -o                           like -l, but do not list group information
  -p, --indicator-style=slash
                               append / indicator to directories
  -q, --hide-control-chars     print ? instead of nongraphic characters
      --show-control-chars     show nongraphic characters as-is (the
default,
                               unless program is 'ls' and output is a
terminal)
  -Q, --quote-name             enclose entry names in double quotes
      --quoting-style=WORD     use quoting style WORD for entry names:
                               literal, locale, shell, shell-always,
                               shell-escape, shell-escape-always, c,
escape
                               (overrides QUOTING_STYLE environment
variable)
  -r, --reverse                reverse order while sorting
  -R, --recursive              list subdirectories recursively
  -s, --size                   print the allocated size of each file, in
blocks
  -S                           sort by file size, largest first
      --sort=WORD              sort by WORD instead of name: none (-U),
size (-S),
                               time (-t), version (-v), extension (-X)
      --time=WORD              change the default of using modification
times;
                               access time (-u): atime, access, use;
                               change time (-c): ctime, status;
                               birth time: birth, creation;
                               with -l, WORD determines which time to show;
                               with --sort=time, sort by WORD (newest
first)
      --time-style=TIME_STYLE  time/date format with -l; see TIME_STYLE
below
```

```
  -t                         sort by time, newest first; see --time
  -T, --tabsize=COLS         assume tab stops at each COLS instead of 8
  -u                         with -lt: sort by, and show, access time;
                               with -l: show access time and sort by
name;
                               otherwise: sort by access time, newest
first
  -U                         do not sort; list entries in directory order
  -v                         natural sort of (version) numbers within
text
  -w, --width=COLS           set output width to COLS.  0 means no limit
  -x                         list entries by lines instead of by columns
  -X                         sort alphabetically by entry extension
  -Z, --context              print any security context of each file
  -1                         list one file per line.  Avoid '\n' with -q
or -b
      --help     display this help and exit
      --version  output version information and exit
...
```

Based on the information provided above, answer the given subquestions.

**Sub questions**

**Question Number : 250 Question Id : 640653563957 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 bash function `cdlmd` that changes the current working directory to the recently modified directory in the current directory.

**Options :**

6406531885203. ✖

```
cdlmd() {
    ls | head -1 | cd
}
```

6406531885204.

```
cdlmd() {
    ls -t -d */ | xargs cd
}
```

6406531885205. ✔

```
cdlmd() {
    ls -t -d */ | head -1 | xargs cd
}
```

6406531885206. ✖

```
cdlmd() {
    ls -g | head -1 | xargs cd
}
```

**Question Number : 251 Question Id : 640653563958 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

Which of the following command prints all the keys present in an bash associative array.

**Options :**

6406531885207. ✖ `$arr`

6406531885208. ✖ `$arr[@]`

6406531885209. ✔ `${!arr[@]}`

6406531885210. ✖ `${arr[@]}`

**Sub-Section Number :** 8

| | |
|---|---|
| **Sub-Section Id :** | 64065380404 |
| **Question Shuffling Allowed :** | No |
| **Is Section Default? :** | null |

**Question Id : 640653563963 Question Type : COMPREHENSION Sub Question Shuffling Allowed : No Group Comprehension Questions : No Question Pattern Type : NonMatrix Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Question Numbers : (252 to 253)**

Question Label : Comprehension

A student named Meena wrote a shell script `exoc.sh` such that everytime `main.py` is changed (change in modification time) the shell script is executed on python script file.

Script: exoc.sh

```
#!/usr/bin/python3

while true; do
        lmt=$(stat -c %Y "$1")
        if [[ "$lmt" == "$plmt" ]]; then
                clear
                echo "[$(date +"%H:%M:%S") STARTED]"
                python3 "$1"
                echo "[$(date +"%H:%M:%S") ENDED]"
        fi
        plmt="$lmt"
        sleep 1
done
```

Execution

```
$ ls -l
total 16K
drwxrwxr-x  2 meena meena 4.0K Apr  9 10:53 ./
drwxr-xr-x 34 meena meena  12K Apr  9 10:52 ../
-rwxrwxr-x  1 meena meena   52 Apr  9 10:52 exoc.sh
-rw-rw-r--  1 meena meena  462 Apr  9 10:53 main.py
$ echo main.py | ./exoc.sh
```

Hints

```
$ stat --help
Usage: stat [OPTION]... FILE...
Display file or file system status.

Mandatory arguments to long options are mandatory for short options too.
  -L, --dereference     follow links
  -f, --file-system     display file system status instead of file status
```

```
      --cached=MODE     specify how to use cached attributes;
                          useful on remote file systems. See MODE below
  -c  --format=FORMAT   use the specified FORMAT instead of the default;
                          output a newline after each use of FORMAT
      --printf=FORMAT   like --format, but interpret backslash escapes,
                          and do not output a mandatory trailing newline;
                          if you want a newline, include \n in FORMAT
  -t, --terse           print the information in terse form
      --help     display this help and exit
      --version  output version information and exit

The --cached MODE argument can be; always, never, or default.
`always` will use cached attributes if available, while
`never` will try to synchronize with the latest attributes, and
`default` will leave it up to the underlying file system.

The valid format sequences for files (without --file-system):

  %a   permission bits in octal (note '#' and '0' printf flags)
  %A   permission bits and file type in human readable form
  %b   number of blocks allocated (see %B)
  %B   the size in bytes of each block reported by %b
  %C   SELinux security context string
  %d   device number in decimal
  %D   device number in hex
  %f   raw mode in hex
  %F   file type
  %g   group ID of owner
  %G   group name of owner
  %h   number of hard links
  %i   inode number
  %m   mount point
  %n   file name
  %N   quoted file name with dereference if symbolic link
  %o   optimal I/O transfer size hint
  %s   total size, in bytes
  %t   major device type in hex, for character/block device special files
  %T   minor device type in hex, for character/block device special files
  %u   user ID of owner
  %U   user name of owner
  %w   time of file birth, human-readable; - if unknown
  %W   time of file birth, seconds since Epoch; 0 if unknown
  %x   time of last access, human-readable
```

```
  %X   time of last access, seconds since Epoch
  %y   time of last data modification, human-readable
  %Y   time of last data modification, seconds since Epoch
  %z   time of last status change, human-readable
  %Z   time of last status change, seconds since Epoch

Valid format sequences for file systems:

  %a   free blocks available to non-superuser
  %b   total data blocks in file system
  %c   total file nodes in file system
  %d   free file nodes in file system
  %f   free blocks in file system
  %i   file system ID in hex
  %l   maximum length of filenames
  %n   file name
  %s   block size (for faster transfers)
  %S   fundamental block size (for block counts)
  %t   file system type in hex
  %T   file system type in human readable form

--terse is equivalent to the following FORMAT:
    %n %s %b %f %u %g %D %i %h %t %T %X %Y %Z %W %o %C
--terse --file-system is equivalent to the following FORMAT:
    %n %i %l %t %s %S %b %f %a %c %d

NOTE: your shell may have its own version of stat, which usually
supersedes
the version described here.  Please refer to your shell's documentation
for details about the options it supports.
```

Based on the above data, answer the given subquestions.

**Sub questions**

**Question Number : 252 Question Id : 640653563964 Question Type : MSQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0**

**Correct Marks : 8 Selectable Option : 0**

Question Label : Multiple Select Question

Apparently, her code had some issues. Identify all the flaws from the following options to facilitate debugging the script and the execution steps.

**Options :**

6406531885231. ✔ Incorrect interpreter in the first line on `exoc.sh`

6406531885232. ✔ Incorrect condition in if statement

6406531885233. ✖ Incorrect assignment to the variable `lmt`

6406531885234. ✖ Incorrect assignment to the variable `plmt`

6406531885235. ✔ Incorrect execution; The proper execution should be `./exoc.sh main.py`

6406531885236. ✖ Incorrect execution; The proper execution should be `exoc.sh main.py`

**Question Number : 253 Question Id : 640653563965 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 outcome of the command `sed -i '/./,$!d' file.txt`?

**Options :**

6406531885237. ✖ Delete all empty lines in the file `file.txt`

6406531885238. ✖

Delete all lines with '.' character in the file `file.txt`

6406531885239. ✔ Delete all empty lines before the first non-empty line in the file `file.txt`

6406531885240. ✖ Delete all empty lines after the last non-empty line in the file `file.txt`

# MLP

| | |
|---|---|
| **Section Id :** | 64065338327 |
| **Section Number :** | 10 |
| **Section type :** | Online |
| **Mandatory or Optional :** | Mandatory |
| **Number of Questions :** | 37 |
| **Number of Questions to be attempted :** | 37 |
| **Section Marks :** | 100 |
| **Display Number Panel :** | Yes |
| **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 :** | 64065380405 |
| **Question Shuffling Allowed :** | No |
| **Is Section Default? :** | null |