System Commands

Section Id :	64065333942
Section Number :	14
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	13
Number of Questions to be attempted :	13
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 :	64065373987
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 230 Question Id : 640653521226 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"

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 :

6406531737270. ✔ YES

6406531737271. ✖ NO

Sub-Section Number :	2
Sub-Section Id :	64065373988
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 231 Question Id : 640653521227 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

```
for i in *; do
    # -d is an unary operator returns exit status 0
    # if the operand is a directory
    if [ -d "$i" ]; then
        mv "$i" "$i.d"
    fi
done
```

Choose the correct statement with respect to the above script.

Options :

6406531737272. ✔ The files in the current directory will not be renamed

6406531737273. ✖ The directories in the subdirectories of the current directory will be renamed

6406531737274. ✖ Only the empty directories will be renamed

6406531737275. ✖ The files are moved from the current directory to another directory with its name suffixed by ".d"

Question Number : 232 Question Id : 640653521235 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

Choose the command to find all occurrences of `{{DATE}}` in the file `template` and replace with the value of shell variable `DATE` in the file. Note that there may be spaces between `{{` and `DATE` and `DATE` and `}}`.
Hint: The option `-i` in SED does the in-place replacement.

Options :

6406531737300. ✖ `sed "s/{{DATE}}/$DATE/g" template`

6406531737301. ✖ `sed -i 's/{{[ ]*DATE[ ]*}}/$DATE/g' template`

6406531737302. ✔ `sed -i "s/{{[ ]*DATE[ ]*}}/$DATE/g" template`

6406531737303. ✖ `sed -i "s/{{[ ]*DATE[ ]*}}/$DATE/" template`

Question Number : 233 Question Id : 640653521236 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 file 'dates.txt' has a list of dates in MM/DD/YYYY format. Which of the following commands can be used to convert it to YYYY-MM-DD format?
Hint: SED uses Basic Regular Expression Engine (BRE) by default.

Options :

6406531737304. ✖ `sed 's/\([0-9]{2}\)\/\([0-9]{2}\)\/\([0-9]{4}\)/\3-\1-\2/' dates.txt`

6406531737305. ✖ `sed 's/([0-9]{2})/([0-9]{2})/([0-9]{4})/\3-\1-\2/' dates.txt`

6406531737306. ✖ `sed 's/\([0-9]\{2\}\)\/\([0-9]\{2\}\)\/\([0-9]\{4\}\)/\4-\2-\2/' dates.txt`

6406531737307. ✔ `sed 's/\([0-9]\{2\}\)\/\([0-9]\{2\}\)\/\([0-9]\{4\}\)/\3-\1-\2/' dates.txt`

Question Number : 234 Question Id : 640653521237 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 quoted CSV file, the fields are bound by double quotes. Given below is an example for quoted CSV file.

```
"Nasrin","Guindy, Chennai","12389"
"Ram Kumar","Daryaganj, Delhi","09890"
```

Note that `Daryaganj, Delhi` is a single field inside the quotes CSV.

Write a SED script to convert the quoted CSV to Tab Separated Value file(TSV) and remove the quotes.

Assume that the field do not contain tabs or a single comma character. The tab character can be represented by `\t`

Options :

6406531737308. ✖
```
s/"/\t/g
```

6406531737309. ✔
```
s/","/\t/g
s/"//g
```

6406531737310. ✖
```
s/"//g
s/","/\t/g
```

6406531737311. ✖
```
s/,/\t/g
s/"//g
```

Question Number : 235 Question Id : 640653521242 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 commands will print the file while prepending the line number to the start of each line, irrespective of the data in the file?

Options :

6406531737328. ✖
```
awk 'END {print NR,$0}' employee_details.txt
```

6406531737329. ✖
```
awk 'BEGIN{FS=","}{print NR,$1}' employee_details.txt
```

6406531737330. ✔
```
awk '{print NR,$0}' employee_details.txt
```

6406531737331. ✖
```
awk '{print $1,$0}' employee_details.txt
```

Sub-Section Number :	3
Sub-Section Id :	64065373989
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 236 Question Id : 640653521228 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

```
for i in *; do

    ** MISSING COMMAND **

    # -d is an unary operator returns exit status 0
    # if the operand is a directory
    if [ -d "$i" ]; then
        mv "$i" "$i.d"
    fi
done
```

Select the missing command to make the above script to produce the same result on every execution. The file/directory names will be the same after the first and every other execution will be the same.

Hint: The option `-q` in grep will not print output only returns the exit status

Options :

6406531737276. ✖ No change required. The given script produces the same result on every execution.

6406531737277. ✖ `ls | grep -q ".d$" && continue`

6406531737278. ✖ `ls | grep -q "\.d$" && continue`

6406531737279. ✔ `echo "$i" | grep -q "\.d$" && continue`

Question Number : 237 Question Id : 640653521238 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

```
awk '
    NR == FNR {
        arr[$0]++
    }
    NR != FNR && !arr[$0] {
        print $0
    }
' file_1 file_2
```

What does the output from the above command represent?

Options :

6406531737312. ✖ Line that present in file_1 and file_2

6406531737313. ✖ Line that present in file_1 but not in file_2

6406531737314. ✔ Line that present in file_2 but not in file_1

6406531737315. ✖ Line that present in file_1 or file_2; the first occurrence will be printed

Question Number : 238 Question Id : 640653521241 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

Here are the top five lines of access log of a server.

```
103.47.219.249 - - [27/Jan/2022:00:01:11 +0530] "GET / HTTP/1.1" 301 429
"-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7
(KHTML, like Gecko) Version/9.1.2 Safari/601.7.7"
54.209.123.136 - - [27/Jan/2022:00:01:18 +0530] "GET
/AlloyOnto/AlloyOnto.owl HTTP/1.1" 301 494 "-" "Python-urllib/3.6"
54.209.123.136 - - [27/Jan/2022:00:01:18 +0530] "GET
/AlloyOnto/AlloyOnto.owl HTTP/1.1" 301 494 "-" "Python-urllib/3.6"
54.209.123.136 - - [27/Jan/2022:00:01:19 +0530] "GET
/AlloyOnto/AlloyOnto.owl HTTP/1.1" 200 1410215 "-" "Python-urllib/3.6"
54.209.123.136 - - [27/Jan/2022:00:01:19 +0530] "GET
/AlloyOnto/AlloyOnto.owl HTTP/1.1" 200 1410215 "-" "Python-urllib/3.6"
```

Given the following AWK script is executed on the access log file. What is the expected output from the AWK script?

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

{
  datetime = $4":"$5
  time=substr(datetime, 14, 8)

  if ( time < "06:00:00" ) {
    if ( $1 in ip ) { ip[$1]++ }
    else { ip[$1]=1 }
  }
}

END {
  mx=0
  for (i in ip) {
    if (ip[i] > mx) {
      mx = ip[i]
      mxip = i
    }
  }
  print mxip
}
```

Options :

6406531737324. ✖ The IP address of the client that made most requests of all time

6406531737325. ✖ The IP address of the client that made the least requests from 6 am to midnight.

6406531737326. ✔ The IP address of the client that made most requests from midnight to 6 am.

6406531737327. ✖ The IP address of the client that made most requests from 6 am to midnight.

Sub-Section Number : 4

Sub-Section Id : 64065373990

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 239 Question Id : 640653521229 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

```
$ xargs --help | head -2
Usage: xargs [OPTION]... COMMAND [INITIAL-ARGS]...
Run COMMAND with arguments INITIAL-ARGS and more arguments read from
input.

$ ls -l
total 20
drwxrwxr-x  3 amit amit  4096 Feb 28 13:29 .
drwxr-xr-x 27 amit amit 12288 Feb 28 13:28 ..
-rw-rw-r--  1 amit amit     0 Feb 28 13:29 a
-rw-rw-r--  1 amit amit     0 Feb 28 13:29 b
-rw-rw-r--  1 amit amit     0 Feb 28 13:29 c
drwxrwxr-x  2 amit amit  4096 Feb 28 13:29 d

$ ls | xargs echo
a b c d
```

Select the command(s) to move the files a, b and c to the directory d in the current working directory.

Options :

6406531737280. ✔ `mv a b c d`

6406531737281. ✔ `ls | xargs mv -t d`

6406531737282. ✔ `ls | sort | xargs mv`

6406531737283. ✔ `mv *`

Question Number : 240 Question Id : 640653521239 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

```
awk '
    {
        arr[$0]++
    }
    END {
        for (i in arr) {
            if (arr[i] > 2) {
                print i
            }
        }
    }
' file_1 file_2 file_3
```

The above command prints a line under which condition?

Options :

6406531737316. ✖ If a line is present once in any two files

6406531737317. ✔ If a line is present in all three files

6406531737318. ✔ If a line is present in any two files and its total occurrence is at least 3

6406531737319. ✖ If a line is present in only one file but its total the occurrence is at most 2

Question Number : 241 Question Id : 640653521240 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

The structure of AWK blocks are provided below

```
pattern { procedure  }
```

Which of the statement(s) are true regarding AWK.

Options :

6406531737320. ✔ BEGIN block will execute the script before reading the file.

6406531737321. ✔ The AWK script that only has a BEGIN block does not require file/stdin.

6406531737322. ✔ END block will execute once all the lines/records from the files/stdin are read.

6406531737323. ✔ The block without any pattern will execute for all the lines/records from files/stdin.

Sub-Section Number :	5
Sub-Section Id :	64065373991
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653521230 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 : (242 to 245)

Question Label : Comprehension

Assume the filenames will not have a colon (:) in it.

Use the console output to answer the given subquestions.

```
$ grep --help
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.
    .......
  -i, --ignore-case         ignore case distinctions in patterns and data
      --no-ignore-case      do not ignore case distinctions (default)
    .......
Output control:
    .......
  -o, --only-matching       show only nonempty parts of lines that match
  -q, --quiet, --silent     suppress all normal output
    .......
  -r, --recursive           like --directories=recurse
    .......
  -c, --count               print only a count of selected lines per FILE
    .......

$ sort --help
Usage: sort [OPTION]... [FILE]...
  or:  sort [OPTION]... --files0-from=F
Write sorted concatenation of all FILE(s) to standard output.

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

Mandatory arguments to long options are mandatory for short options too.
Ordering options:

    .......
  -n, --numeric-sort          compare according to string numerical value
    .......
  -r, --reverse               reverse the result of comparisons
    .......
  -C, --check=quiet, --check=silent  like -c, but do not report first bad
line
      --compress-program=PROG  compress temporaries with PROG;
                               decompress them with PROG -d
      --debug               annotate the part of the line used to sort,
                               and warn about questionable usage to stderr
      --files0-from=F       read input from the files specified by
                            NUL-terminated names in file F;
```

```
                           If F is - then read names from standard input
  -k, --key=KEYDEF         sort via a key; KEYDEF gives location and
type
    .......
  -t, --field-separator=SEP  use SEP instead of non-blank to blank
transition
    .......

$ grep -ric "print"
tools/example.sh:0
tools/upgrade.sh:12
tools/uninstall.sh:12
tools/install.sh:44
tools/autossh.sh:3

$ cat data
13118,21233,24423
29515,22595,27723
20753,2195,4761
29399,23451,23061
725,11432,26480

$ cat data | sort -t , -k 3 -n
20753,2195,4761
29399,23451,23061
13118,21233,24423
725,11432,26480
29515,22595,27723
```

Sub questions

Question Number : 242 Question Id : 640653521231 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

```
$ grep -ric "print" |
    sort -nr -t : -k 2 |
    cut -d: -f1 |
    head -n 10
```

What will be the output from the above command represent?

Options :

6406531737284. ✖ Total number of *lines* have the string "print" in all files in the current and subdirectories

6406531737285. ✖ Total number of occurrences of "print" in all files in the current and subdirectories

6406531737286. ✔ Top 10 files that contains most number of lines have "print" among all files in the current and subdirectories

6406531737287. ✖ Top 10 files that contains the least number of lines have "print" among all files in the current and subdirectories

Question Number : 243 Question Id : 640653521232 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

```
$ grep -ric "print" |
    cut -d: -f2 |
    while read n; do
        count=${count:-0}
        count=$((count + n))
        echo $count
    done | tail -n 1
```

What will be the output from the above command represent?

Options :

6406531737288. ✔ Total number of *lines* have the string "print" in all files in the current and subdirectories

6406531737289. ✖ Total number of *files* have the string "print" in the current and subdirectories

6406531737290. ✖ Total number of *occurrences* of "print" in all files in the current and subdirectories

6406531737291. ✖ Total number of *occurrences* of "print" in all files in the current directory

Question Number : 244 Question Id : 640653521233 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

```
$ grep -ric "print" |
    cut -d: -f2 |
    while read n; do
        count=${count:-0}
        count=$((count + n))
        echo $count
    done | tail -n 1
```

What will be the equivalent command(s) using AWK with respect to the provided data?

Options :

6406531737292. ✔

```
grep -ric "print" |
    cut -d: -f2 |
    awk '{c+=$1} END{print c}'
```

6406531737293. ✔

```
grep -ric "print" |
    awk 'BEGIN{FS=":"} {c+=$2} END{print c}'
```

6406531737294. ✖

```
grep -ric "print" |
    awk '{c+=$1} END{print c}'
```

6406531737295. ✖

```
grep -ric "print" |
    awk 'BEGIN{FS=":"} {c=$2} END{print c}'
```

Question Number : 245 Question Id : 640653521234 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

```
$ grep -ric "print" |
    cut -d: -f2
```

What will be the equivalent command(s) using SED with respect to the provided data?

Options :

6406531737296. ✔

```
grep -ric "print" |
    sed 's/.*://'
    # Assumption: no colon in the filename
    # refer sample output
```

6406531737297. ✔

```
grep -ric "print" |
    sed 's/[^:]*://'
```

6406531737298. ✖

```
grep -ric "print" |
    sed 's/[^:]*:[^:]*//g'
```

6406531737299. ✖

```
grep -ric "print" |
    sed 's/:.*//'
```