Question Number : 263 Question Id : 640653445714 Question Type : MCQ Is Question Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 1

Question Label : Multiple Choice Question

What is called as efficiency?

Options :

6406531485085. ✖ Output/(1-input)

6406531485086. ✖ 1 - (output/input)

6406531485087. ✔ Output/Input

6406531485088. ✖ None of these

System commands

Section Id :	64065328989
Section Number :	15
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	12
Number of Questions to be attempted :	12
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 :	64065363376
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 264 Question Id : 640653445715 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 :

6406531485089. ✔ Yes

6406531485090. ✖ No

Sub-Section Number :	2
Sub-Section Id :	64065363377
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 265 Question Id : 640653445716 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 of the given command?

```
awk '/^[a-zA-Z]/ { c++ } END{ print c }' myfile.txt
```

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 :

14

Sub-Section Number :	3
Sub-Section Id :	64065363378
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 266 Question Id : 640653445717 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 below script?
Note that a single negative argument to `seq` will not give any output.

```
for i in $(seq 9); do
    for j in $(seq $((5-i))); do
        echo -n '* '
    done
    for j in $(seq $((i-5))); do
        echo -n '* '
    done
    echo
done
```

Options :

6406531485092. ✔

```
* * * *
* * *
* *
*

*
* *
* * *
* * * *
```

6406531485093. ✖

```
* * * * *
* * * *
* * *
* *
*
* *
* * *
* * * *
* * * * *
```

6406531485094. ✖

```
*
* *
* * *
* * * *
* * * * *
* * * * * *
* * * * * * *
* * * * * * * *
* * * * * * * * *
```

```
* * * * * * * * * *
* * * * * * * * *
* * * * * * * *
* * * * * *
* * * * *
* * * *
* * *
* *
*
```

6406531485095. ✖

Question Number : 267 Question Id : 640653445720 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

How many lines will be printed if the following command is executed? Assume that ***myfile.txt*** contains more than 3 lines.

```
sed '
1 i ---START---
3 c ---THREE---
$ a ---END---
' myfile.txt
```

Options :

6406531485104. ✖ Same as the number of lines in *myfile.txt*

6406531485105. ✖ Number of lines in *myfile.txt* + 1

6406531485106. ✔ Number of lines in *myfile.txt* + 2

6406531485107. ✖ Number of lines in *myfile.txt* + 3

Question Number : 268 Question Id : 640653445721 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 that converts comma separated file named *data.csv* to a tab separated file named *data.tsv* Assume there is no comma in the field values.

Options :

6406531485108. ✖ `mv data.csv data.tsv`

6406531485109. ✖ `sed 's/\t/,/' data.tsv > data.csv`

6406531485110. ✔ `sed 's/,/\t/g' data.csv > data.tsv`

6406531485111. ✖ `sed '/,/ c \t' data.csv > data.tsv`

Sub-Section Number :	4
Sub-Section Id :	64065363379
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 269 Question Id : 640653445718 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

Select the command(s) that prints only the lines containing the string `TODO` in any part of the line in the file `todo.txt`

Options :

6406531485096. ✖ `awk '{print TODO}' todo.txt`

6406531485097. ✔ `awk '/TODO/ {print}' todo.txt`

6406531485098. ✖ `awk '{ if ($1 ~ /TODO/) {print;} }' todo.txt`

6406531485099. ✔ `awk '{ if ($0 ~ /TODO/) {print;} }' todo.txt`

Sub-Section Number :	5
Sub-Section Id :	64065363380
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 270 Question Id : 640653445719 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

Which command will print only the multi-line strings from the python file named ***myscript.py***?

Example:

```
my_multiline = """
a
b
c
d
"""
```

Note:

1. The multi-line string will start with """ and ends with """
2. There will be a single equal sign (=) before starting """ and there can be spaces in-between them.
3. There is no text after starting """
4. There is no text before and after at ending """ in the same line.
5. -v option in grep will print only the non-matched lines.

Options :

6406531485100. ✔ `sed -n '/= *"""/,/"""/ p' myscript.py | grep -v '"""'`

6406531485101. ✖ `sed '/= *"""/,/"""/ d' myscript.py`

6406531485102. ✖ `sed -n '/= *"""/,/"""/ p' myscript.py | grep '"""'`

6406531485103. ✖ `sed -n '/=*"""/,/"""/ p' myscript.py | grep """`

Sub-Section Number : 6

Sub-Section Id : 64065363381

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 271 Question Id : 640653445722 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

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

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

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

Options :

6406531485112. ✔ `sed -nE '/.+:.:[[:digit:]]{4,}:.*bash/ p' /etc/passwd`

6406531485113. ✖ `sed -nE '/.+:.:[[:digit:]]{3}:.*bash/ p' /etc/passwd`

6406531485114. ✔ `awk '$3 > 999 && $7 ~ /.*bash/ {print $1}' /etc/passwd`

6406531485115. ✖ `awk 'BEGIN{FS=":"} $3 > 999 && $7 ~ /.*bash/ {print $1}' /etc/passwd`

Question Number : 272 Question Id : 640653445723 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 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 executing the below script.

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

Options :

6406531485116. ✖ file0

6406531485117. ✔ file1

6406531485118. ✖ file2

6406531485119. ✔ file3

6406531485120. ✔ file4

Question Number : 273 Question Id : 640653445728 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

Consider the below information to answer the question.

```
$ whatis kill
kill (1)              - send a signal to a process
kill (2)              - send signal to a process

$ kill --help
kill: kill [-s sigspec | -n signum | -sigspec] pid | jobspec ... or kill
-l [sigspec]
    Send a signal to a job.

    Send the processes identified by PID or JOBSPEC the signal named by
    SIGSPEC or SIGNUM.  If neither SIGSPEC nor SIGNUM is present, then
    SIGTERM is assumed.

    Options:
      -s sig    SIG is a signal name
      -n sig    SIG is a signal number
      -l        list the signal names; if arguments follow `-l` they are
                assumed to be signal numbers for which names should be
listed
      -L        synonym for -l


    Kill is a shell builtin for two reasons: it allows job IDs to be
used
    instead of process IDs, and allows processes to be killed if the
limit
    on processes that you can create is reached.

    Exit Status:
    Returns success unless an invalid option is given or an error
occurs.

$ whatis killall
killall (1)           - kill processes by name

$ killall --help
Usage: killall [OPTION]... [--] NAME...
       killall -l, --list
       killall -V, --version

  -e,--exact           require exact match for very long names
  -I,--ignore-case     case insensitive process name match
  -g,--process-group   kill process group instead of process
  -y,--younger-than    kill processes younger than TIME
  -o,--older-than      kill processes older than TIME

   -i,--interactive    ask for confirmation before killing
   -l,--list           list all known signal names
   -q,--quiet          dont print complaints
   -r,--regexp         interpret NAME as an extended regular expression
   -s,--signal SIGNAL  send this signal instead of SIGTERM
   -u,--user USER      kill only process(es) running as USER
   -v,--verbose        report if the signal was successfully sent
   -V,--version        display version information
   -w,--wait           wait for processes to die
   -n,--ns PID         match processes that belong to the same namespaces
                       as PID
   -Z,--context REGEXP kill only process(es) having context
                       (must precede other arguments)
```

Select the bash script(s) that kills all the processes of *sleep*.

Options :

6406531485135. ✔
```
killall sleep
```

6406531485136. ✖
```
while (ps | grep sleep); do
        kill sleep
done
```

```
while (ps | grep sleep); do
        kill $(ps | grep sleep | head -1 | awk '{print $1}')
done
```

6406531485137. ✔

```
while (ps | grep sleep); do
        kill $(ps | head -1 | awk '{print $1}')
done
```

6406531485138. ✖

Sub-Section Number :	7
Sub-Section Id :	64065363382
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653445724 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 : (274 to 276)

Question Label : Comprehension

The file */etc/group* stores the group information of the system in the format given below

```
GroupName:x:GID:Members(separated by comma)
```

An example line from */etc/group* file is given below

```
student:x:214:ram,ahmed,robert,seema
```

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
```

From `man awk`,

```
split(s, a [, r [, seps] ])
  Split  the string s into the array a and the separators array
  seps on the regular expression r, and return  the  number  of
  fields.   If  r is omitted, FS is used instead.  The arrays a
  and seps are cleared first.  seps[i] is the  field  separator
  matched  by  r  between  a[i]  and  a[i+1].  If r is a single
  space, then leading whitespace in s goes into the extra array
  element  seps[0]  and trailing whitespace goes into the extra
  array element  seps[n],  where  n  is  the  return  value  of
  split(s, a, r, seps).  Splitting behaves identically to field
  splitting, described above.  In particular, if r is a single-
  character  string, that string acts as the separator, even if
  it happens to be a regular expression metacharacter.
```

script_1

```
awk '
BEGIN {
    FS=":"
    A=0
    B=""
    C=""
}
{
    n = split($4, arr, ",")
    if (n > A) {
        A = n
        B = $1
        C = $4
    }
}
END {
    print A,B,C # Line 17
}
' /etc/group
```

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 274 Question Id : 640653445725 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 value of `A` when the print statement in line 17 of the given script is executed?

Options :

6406531485121. ✖ The number of lines in the input file

6406531485122. ✖ The number of groups having more than one users

6406531485123. ✔ The maximum number of users in any group

6406531485124. ✖ The minimum number of users in any group

Question Number : 275 Question Id : 640653445726 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 value of `B` when the print statement in line 17 of the given script is executed?

Options :

6406531485125. ✖ The first field of last line

6406531485126. ✔ The last occurrence of group with the maximum number of users

6406531485127. ✖ The last occurrence of group with the minimum number of users

6406531485128. ✖ The last occurrence of group with no users

Question Number : 276 Question Id : 640653445727 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 if the output from the script ***script_1*** is piped to the script below?

```
awk '
{
    split($3, arr, ",")
    for (i in arr) {

        # Execute the string and get the first line of
        # the output in the variable u
        "grep ^"arr[i]": /etc/passwd" | getline u

        split(u, arr2, ":")
        print arr2[3]
    }
}
'
```

Options :

6406531485129. ✖ The other group names in which the users from the largest group are present.

6406531485130. ✖ The other group names in which the users from the smallest non-zero member group are present.

6406531485131. ✖ The GIDs of the users belonging to the largest group.

6406531485132. ✖ The GIDs of the users belonging to the smallest group.

6406531485133. ✔ The UIDs of the users belonging to the largest group.

6406531485134. ✖ The UIDs of the users belonging to the smallest group.

Sub-Section Number :	8
Sub-Section Id :	64065363383
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Id : 640653445729 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 : (277 to 279)

Question Label : Comprehension

In a computer science project, the students are asked to provide their remote git repository URLs. The URLs are then stored in the file named `data.csv`. There is a bash script written to validate the programs(which they submitted as git repository) written by the students. The project was structured such that `main.sh` in the project's root directory takes standard input and provides the standard output that is used for validation.

Note:

1. The fields of `data.csv` are roll number and repository URL respectively.
2. The test cases are located in the directory `testcases` which is located in home directory.
3. The structure of `testcases` directory is given below

```
~/testcases/
├── 1/
│   ├── input.txt
│   └── output.txt
├── 2/
│   ├── input.txt
│   └── output.txt
├── 3/
│   ├── input.txt
│   └── output.txt
..
```

4. All the required packages (git, diff, ...) are already installed in the system.

```bash
#!/bin/bash

# PART-1 Getting the project files using git
# Input Field Separator (IFS)
IFS=","  # Line 1-a
while read rollno url; do # Line 1-b
    # Clone files to a new directory named with rollno
    git clone $url $rollno # Line 1-c
done  # Line 1-d

# ********************************************

# PART-2 Evaluating the project using test cases
# and generate a log file with evaluation results
TESTCASE_DIR=~/testcases # Line 2-a
LOG_FILE=~/log.csv; echo "" > $LOG_FILE
while read rollno url; do # Line 2-c
    cd $rollno # Line 2-d
    for tc in $TESTCASE_DIR/*; do
        bash main.sh < $tc/input.txt > /tmp/tmp_output # Line 2-f
        diff $tc/output.txt /tmp/tmp_output > /dev/null 2>&1 # Line 2-g
        if [ $? == 0 ]; then # Line 2-h
            echo "$rollno,$tc,PASS" >> $LOG_FILE # Line 2-i
        else
            echo "$rollno,$tc,FAIL" > $LOG_FILE # Line 2-k
        fi
    done
    cd ..
done < data.csv

# ********************************************

# PART-3 Generating results to terminal
TOTAL_TESTCASES=$(ls $TESTCASE_DIR | wc -l) # Line 3-a
echo "SUMMARY"
while read rollno url; do # Line 3-c
    passed_tc=$(grep PASS $LOG_FILE | wc -l) # Line 3-d
    echo "$rollno $passed_tc/$TOTAL_TESTCASES" # Line 3-e
done < data.csv
```

Based on the above data, answer the given subquestions.

Sub questions

Question Number : 277 Question Id : 640653445730 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

Identify all the mistakes in the PART-1 of the script.

Options :

6406531485139. ✖ Line 1-a: The value of the IFS variable is wrong

6406531485140. ✔ Line 1-b: No standard input is provided, thus replacing Line 1-a with `cat data.txt | while read rollno url; do` will resolve the issue.

6406531485141. ✖ Line 1-c: `git clone $url $rollno` is an invalid statement because no command named git is installed in the system.

6406531485142. ✔ Line 1-d: No standard input is provided, thus replacing this with `done < data.txt` will resolve the issue.

Question Number : 278 Question Id : 640653445731 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

Read the description given in the comments and identify all the mistakes in the PART- 2 of the script.

Options :

6406531485143. ✖ Line 2-c: Standard input not provided

6406531485144. ✖ Line 2-f: Incorrect IO redirection

6406531485145. ✖ Line 2-g: Incorrect IO redirection

6406531485146. ✖ Line 2-h: Incorrect condition

6406531485147. ✖ Line 2-i: Incorrect IO redirection

6406531485148. ✔ Line 2-k: Incorrect IO redirection

Question Number : 279 Question Id : 640653445732 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

Read the description given in the comments and identify all the mistakes in the PART- 3 of the script.

Options :

6406531485149. ✖ Line 3-a: Incorrect calculation of the total test cases

6406531485150. ✖ Line 3-c: Standard input not provided

6406531485151. ✔ Line 3-d: Incorrect calculation of the passed test cases, it should be

```
passed_tc=$(grep "^$rollno," $LOG_FILE | grep PASS | wc -l)
```

6406531485152. ✖ Line 3-e: Some of the referred variables are not defined