

Answers Type : Range

Text Areas : PlainText

Possible Answers :

0.4 to 0.42

Question Number : 234 **Question Id :** 640653577988 **Question Type :** MSQ **Is Question**

Mandatory : No **Calculator :** None **Response Time :** N.A **Think Time :** N.A **Minimum Instruction Time :** 0

Correct Marks : 1 **Max. Selectable Options :** 0

Question Label : Multiple Select Question

Given the computed elasticity, what can you say about the “Butter Milk” demand? (choose all that are applicable)

Options :

6406531930111. ✖ It is elastic

6406531930112. ✔ It is inelastic

6406531930113. ✖ It is a luxury

6406531930114. ✖ It is a necessity

6406531930115. ✖ Cannot say, insufficient information

System Commands

Section Id :	64065339080
Section Number :	15
Section type :	Online
Mandatory or Optional :	Mandatory
Number of Questions :	16
Number of Questions to be attempted :	16
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 :	64065382637
Question Shuffling Allowed :	No
Is Section Default? :	null

Question Number : 235 Question Id : 640653577992 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 :

6406531930125. ✓ YES

6406531930126. ✗ NO

Sub-Section Number :	2
Sub-Section Id :	64065382638
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 236 Question Id : 640653577993 Question Type : MCQ Is Question

Mandatory : No Calculator : None Response Time : N.A Think Time : N.A Minimum Instruction Time : 0

Correct Marks : 4

Question Label : Multiple Choice Question

```
mkdir mydir
touch mydir/file1.txt;touch mydir/file2.txt
mkdir mydir/subdir;touch mydir/subdir/file3.txt
rmdir mydir/subdir
```

Select the output of the above script.

Options :

6406531930127. ✖ remove directory `subdir`

6406531930128. ✖ remove directory `dir` and `subdir`

6406531930129. ✖ remove files from `subdir`

6406531930130. ✔ exit with exit code (not 0)

Sub-Section Number : 3

Sub-Section Id : 64065382639

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 237 Question Id : 640653577994 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

```
#!/bin/bash
export MY_VAR="Sunset"
my_var="Sunrise"
echo "Var11 is: $MY_VAR"
echo "Var12 is: $my_var"
(
    my_var="Sun_rise"
    MY_VAR_var="Sun_set"
    echo "Var21 is: $MY_VAR"
    echo "Var22 is: $my_var"
    (
        MY_VAR="Sun@rise"
        my_var="Sun@set"
        echo "Var31 is: $MY_VAR"
        echo "Var32 is:${my_var}"
    ) & disown
)
wait
echo "Var41 is: $MY_VAR"
echo "var42 is: $my_var"
```

What will be the value of `Var42` at the end of execution?
Use the following information if needed.

disown: disown [-h] [-ar] [jobspec ... | pid ...]

Remove jobs from current shell.

Removes each JOBSPEC argument from the table of active jobs. Without any JOBSPECs, the shell uses its notion of the current job.

Options:

-a	remove all jobs if JOBSPEC is not supplied
-h	mark each JOBSPEC so that SIGHUP is not sent to the job if the shell receives a SIGHUP
-r	remove only running jobs

Exit Status:

Returns success unless an invalid option or JOBSPEC is given.

wait: wait [-n] [id ...]

Wait for job completion and return exit status.

Waits for each process identified by an ID, which may be a process ID or a job specification, and reports its termination status. If ID is not given, waits for all currently active child processes, and the return status is zero. If ID is a job specification, waits for all processes in that job's pipeline.

If the -n option is supplied, waits for the next job to terminate and returns its exit status.

Exit Status:

Returns the status of the last ID; fails if ID is invalid or an invalid option is given.

Options :

6406531930131. ✓ Sunrise

6406531930132. ✗ Sun_rise

6406531930133. ✗ Sun@rise

6406531930134. ✗ Sun@set

Question Number : 238 Question Id : 640653577995 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

Identify the correct pair for the output of each command.

```
$ date
Mon Jun 12 12:28:02 IST 2023
$ date_disp=$(date) # creates an array. each space
separated fields becomes an element.
$ echo $date_disp
$ echo $date_disp[5]
$ echo ${date_disp[5]}
$ echo ${date_disp:1:8}the
```

Sr.No	command	option	output
1	echo \$date_disp	a	onthe
2	echo \${date_disp[5]}	b	mon[5]
3	echo \$date_disp[5]	c	2023
4	echo \${date_disp:1:8}the	d	Mon

Options :

6406531930135. ✖ 1->b,2->a,3->c,4->d

6406531930136. ✖ 1->d, 2->b, 3 ->c, 4->a

6406531930137. ✔ 1->d, 2->c, 3->b, 4->a

6406531930138. ✖ none of these

Question Number : 239 Question Id : 640653578002 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 lines are a part of a shell script for job submission to a server. Assume that the PBS_JOBID stores 123456.pbs value and \$HOME stores /user value. What would be the value of tempdir variable.

```
tpdir=`echo $PBS_JOBID | cut -f 1 -d .`  
tempdir=$HOME/scratch/job$tpdir  
mkdir -p $tempdir  
cd $tempdir
```

Options :

6406531930160. ✖ /user/scratch/jobpbs

6406531930161. ✖ job123456

6406531930162. ✔ /user/scratch/job123456

6406531930163. ✖ /scratch/jobpbs

Question Number : 240 Question Id : 640653578004 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 most appropriate regex to match an email address. The regex is provided in Extended Regular Expression Engine (ERE).

Options :

6406531930165. ✖ [A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+.[A-Za-z]{2,}

6406531930166.

✓ \b[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}\b

6406531930167. ✖ [A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}

6406531930168. ✖ \b[A-Za-z0-9._%+-]*@[A-Za-z0-9.-]*\.[A-Za-z]{2,}\b

Question Number : 241 Question Id : 640653578005 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


```
$ tr --help
```

```
Usage: tr [OPTION]... SET1 [SET2]
```

Translate, squeeze, and/or delete characters from standard input,
writing to standard output.

-c, -C, --complement	use the complement of SET1
-d, --delete	delete characters in SET1, do
not translate	
-s, --squeeze-repeats	replace each sequence of a
repeated character	
	that is listed in the last
specified SET,	
	with a single occurrence of
that character	
-t, --truncate-set1	first truncate SET1 to length of
SET2	
--help	display this help and exit
--version	output version information and exit

SETs are specified as strings of characters. Most represent themselves.

Interpreted sequences are:

<code>\NNN</code>	character with octal value NNN (1 to 3 octal digits)
<code>\\</code>	backslash
<code>\a</code>	audible BEL
<code>\b</code>	backspace
<code>\f</code>	form feed
<code>\n</code>	new line
<code>\r</code>	return
<code>\t</code>	horizontal tab
<code>\v</code>	vertical tab
<code>CHAR1-CHAR2</code>	all characters from CHAR1 to CHAR2 in ascending order
<code>[CHAR*]</code>	in SET2, copies of CHAR until length of SET1
<code>[CHAR*REPEAT]</code>	REPEAT copies of CHAR, REPEAT octal if starting with 0
<code>[:alnum:]</code>	all letters and digits
<code>[:alpha:]</code>	all letters
<code>[:blank:]</code>	all horizontal whitespace
<code>[:cntrl:]</code>	all control characters
<code>[:digit:]</code>	all digits
<code>[:graph:]</code>	all printable characters, not including space
<code>[:lower:]</code>	all lower case letters
<code>[:print:]</code>	all printable characters, including space
<code>[:punct:]</code>	all punctuation characters
<code>[:space:]</code>	all horizontal or vertical whitespace
<code>[:upper:]</code>	all upper case letters
<code>[:xdigit:]</code>	all hexadecimal digits
<code>[=CHAR=]</code>	all characters which are equivalent to

CHAR

Translation occurs if -d is not given and both SET1 and SET2 appear.

-t may be used only when translating. SET2 is extended to length of

SET1 by repeating its last character as necessary. Excess characters

of SET2 are ignored. Only [:lower:] and [:upper:] are guaranteed to

expand in ascending order; used in SET2 while translating, they may

only be used in pairs to specify case conversion. -s uses the last

specified SET, and occurs after translation or deletion.

GNU coreutils online help:

<<https://www.gnu.org/software/coreutils/>>

Full documentation

<<https://www.gnu.org/software/coreutils/tr>>

or available locally via: info '(coreutils) tr invocation'

Using the above context, choose the command that deletes all occurrences of `a`.

Options :

6406531930169. ✘ `cat myfile.txt | tr 'a' ' ' # there is space between single quotes`

6406531930170. ✘ `cat myfile.txt | tr ' ' 'a' # there is space between single quotes`

6406531930171. ✔ `cat myfile.txt | tr -d 'a'`

6406531930172. ✘ `cat myfile.txt | tr -d '\a'`

Sub-Section Number : 4
Sub-Section Id : 64065382640
Question Shuffling Allowed : Yes
Is Section Default? : null

Question Number : 242 Question Id : 640653577996 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 script takes list of file names as input and finds if the file is executable and counts and lists out executable files. Identify the missing code block from the following options.

```
#!/bin/bash

files=("$@")
executable_files=()

*****
***Missing Code***
*****

echo "Number of executable files: ${#executable_files[@]}"
echo "Executable files:"
for file in "${executable_files[@]}"
do
    echo "$file"
done
```

Options :

6406531930139. ✓

```
for file in "${files[@]}"
do
    if [ -x "$file" ]
    then
        executable_files+=("$file")
    fi
done
```

```
for file in "${files[#]}"
do
    if [ -x "$file" ]
    then
        executable_files+=("$file")
    fi
done
```

6406531930140. ✖

```
for file in "${files[@]}"
do
    if [ -x "$file" ]
    then
        executable_files+=("$file") + 1
    fi
done
```

6406531930141. ✖

```
for file in "${files[@]}"
do
    if [ -f "$file" ]
    then
        executable_files+=("$file")
    fi
done
```

6406531930142. ✖

Correct Marks : 7

Question Label : Multiple Choice Question

Which of the following command will generate output for both `test.out` and `test.err` files. Assume that the `ech` is not a command and will generate a error message.

Options :

6406531930151. ✖ `(echo "test" && ech) > test.err 2>&1 | tee -a test.out`

6406531930152. ✔ `(echo "test" && ech) 2> test.err | tee >test.out`

6406531930153. ✖ `(echo "test" && ech) > test.err 2>&1 | tee >test.out`

6406531930154. ✖ `$(echo "test" && ech) > test.err 2>&1 | tee >test.out`

Sub-Section Number : 5

Sub-Section Id : 64065382641

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 244 Question Id : 640653577997 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 regex to extract only the value of "email" from the below JSON file names as test.json.

```
[
  {
    "name": "Chadwick Cummings",
    "email": "nulla.dignissim.maecenas@hotmail.org",
    "address": "326-2072 Sagittis Road",
    "numberrange": 9,
    "alphanumeric": "ESM17JCJ7NR"
  },
  {
    "name": "Isaac Whitaker",
    "email": "vitae.semper.egestas@icloud.ca",
    "address": "589-9277 Vivamus St.",
    "numberrange": 3,
    "alphanumeric": "CTI05YDP7BX"
  },
  {
    "name": "Bethany Potter",
    "email": "enim.gravida@protonmail.com",
    "address": "P.O. Box 807, 2790 Ut, Ave",
    "numberrange": 7,
    "alphanumeric": "VOE77ZLE00J"
  }
]
```

Hint: `-o` option in `grep` prints only the matched regular expression.

```
grep -o "name" test.json
name
name
name
```

Options :

6406531930143. ✓ `grep "email" test.json | cut -d '"' -f 4`

6406531930144. ✓ `grep -o -E '"email": "[^"]+"' test.json | cut -d '"' -f 4`

6406531930145. ✗ `grep '"email":' test.json | cut -d ' ' -f 2`

6406531930146. ✗ `grep 'email: ' test.json | cut -d '"' -f 4`

Question Number : 245 Question Id : 640653577998 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

Following file contains the information on the websites visited on certain server. From the options identify the correct regular expression (ERE) which can capture the website domain names (<https://www.something.something>, or <http://www.something.something>).

Website	IP Address	Hits	Server Location
https://www.bechtelar.com/alias-similique-ratione-voluptates-aliquam-delectus-gui-cumque-aut	204.83.121.207	0	CK
http://www.okeefe.info/quis-repudiandae-nobis-repellendus-omnis-dolor	78.86.32.75	6	SE
http://macejkovic.com/aut-gui-nostrum-numquam.html	74.58.20.242	1	MU
http://www.cummings.net/magnam-excepturi-eos-rerum-dolores	82.204.55.211	8	PH
http://www.barton.biz/iure-iusto-explicabo-est-soluta-recusandae	252.194.135.149	1	SO
http://www.haag.com/et-exercitationem-id-sunt-sed-laboriosam	1.118.48.149	4	PW
http://www.beer.com/voluptatem-quod-nesciunt-aut.html	71.88.92.193	9	SJ
http://www.mann.org/omnis-ex-in-est-et.html	249.210.50.146	9	BZ
http://towne.com/	58.17.4.75	9	WF
http://www.feeney.net/enim-animi-sapiente-porro-aut-velit-dicta	201.58.189.12	3	ML
http://heaney.com/aliquid-et-rerum-porro-nesciunt-voluptate-quo-sint	9.24.3.149	7	KM
http://bode.com/sint-ut-et-possimus-odit-debitis.html	210.73.162.76	6	BQ

Options :

6406531930147. ✖ `https?://[a-zA-Z]+\.[a-zA-Z]{2,}`

6406531930148. ✔ `http.*//[^\s/]+`

6406531930149. ✔ `https?://[^\s/]+`

6406531930150. ✖ [https://\[a-zA-Z\]+\.\[a-zA-Z\]+\.\[a-zA-Z\]{2,}](https://[a-zA-Z]+\.[a-zA-Z]+\.[a-zA-Z]{2,})

Question Number : 246 Question Id : 640653578001 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

Given the following input identify which of the statements present in options are TRUE.

```
$ ls -li
```

```
23 -rwxr-xr-x 3 root root 35K Jul 4 2019 bunzip2
31611 -rwxr-xr-x 1 root root 2.0M Nov 25 2021 busybox
23 -rwxr-xr-x 3 root root 35K Jul 4 2019 bzip2
25 lrwxrwxrwx 1 root root 6 Jul 4 2019 bzcmp ->
bzdiff
26 -rwxr-xr-x 1 root root 2.1K Jul 4 2019 bzdiff
27 lrwxrwxrwx 1 root root 6 Jul 4 2019 bzegz ->
bzgr
28 -rwxr-xr-x 1 root root 4.8K Jul 4 2019 bzexe
29 lrwxrwxrwx 1 root root 6 Jul 4 2019 bzfgz ->
bzgr
30 -rwxr-xr-x 1 root root 3.6K Jul 4 2019 bzgrep
23 -rwxr-xr-x 3 root root 35K Jul 4 2019 bzip2
31 -rwxr-xr-x 1 root root 14K Jul 4 2019
bzip2recover
32 lrwxrwxrwx 1 root root 6 Jul 4 2019 bz1 -> bzm
33 -rwxr-xr-x 1 root root 1.3K Jul 4 2019 bzmore
```

Options :

6406531930156. ✖ files bzegz and bzfgz are hard links

6406531930157. ✔ files bzegz and bzfgz are soft links

6406531930158. ✖ files bz1 and bzm are hard links

6406531930159. ✔ files bzcac and bzip2 are hard links

Question Number : 247 Question Id : 640653578007 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

```
#!/bin/bash
```

```
directory="mydirectory"
```

```
zip_dest="myarchive.zip"
```

```
read password
```

```
zip -r -P $password $zip_dest $directory
```

Hint:

```
$ man zip
```

```
ZIP(1)                                General Commands Manual
```

```
ZIP(1)
```

NAME

zip - package and compress (archive) files

SYNOPSIS

```
zip [-aABcdDeEfFghjklLmoqrRSTuvVwXyz!@$] [--  
longoption ...] [-b path]  
[-n suffixes] [-t date] [-tt date] [zipfile [file  
...]] [-xi list]
```

...

DESCRIPTION

zip is a compression and file packaging utility for Unix, VMS, MSDOS,

OS/2, Windows 9x/NT/XP, Minix, Atari, Macintosh, Amiga, and Acorn RISC

OS. It is analogous to a combination of the Unix commands tar(1) and

compress(1) and is compatible with PKZIP (Phil Katz's ZIP for MSDOS systems).

...

-P password

--password password

Use password to encrypt zipfile entries (if any). THIS IS INSECURE! Many multi-user operating systems provide ways for any user to see the current command line of any other user; even on stand-alone systems there is always the threat of over-the-shoulder peeking. Storing the plaintext password as part of a command line in an automated script is even worse. Whenever possible, use the non-echoing, interactive prompt to enter passwords. (And where security is truly important, use strong encryption such as Pretty Good Privacy instead of the relatively weak standard encryption provided by zipfile utilities.)

...

`-r`

`--recurse-paths`

Travel the directory structure recursively;

for example:

```
zip -r foo.zip foo
```

or more concisely

```
zip -r foo foo
```

In this case, all the files and directories in foo are saved in

a zip archive named foo.zip, including files with names starting

with ".", since the recursion does not use the shell's file-name

substitution mechanism. If you wish to include only a specific

subset of the files in directory foo and its subdirectories, use

the `-i` option to specify the pattern of files to be included.

You should not use `-r` with the name `"*"`, since that matches

`".."` which will attempt to zip up the parent directory (proba-

bly not what was intended).

Multiple source directories are allowed as

```
in

    zip -r foo foo1 foo2

    which first zips up foo1 and then foo2,
going down each direc-
    tory.

    Note that while wildcards to -r are
typically resolved while re-
    cursing down directories in the file
system, any -R, -x, and -i
    wildcards are applied to internal archive
pathnames once the di-
    rectories are scanned. To have wildcards
apply to files in sub-
    directories when recursing on Unix and
similar systems where the
    shell does wildcard substitution, either
escape all wildcards or
    put all arguments with wildcards in quotes.
This lets zip see
    the wildcards and match files in
subdirectories using them as it
    recurses.

...
```

For the script run.sh, identity the true statement(s) from the following options.

Options :

6406531930174. ✓ The Bash interpreter used to run the run.sh file

6406531930175. ✗ A new directory is created at the end of the execution

6406531930176. ✓ The zip file is password protected

6406531930177. ✗ The password for the zip is "password"

6406531930178. ✓ The password for the zip cannot be obtained from the given script because it is read from the standard input

6406531930179. ✓ A new file, `myarchive.zip` is created at the end of the execution

6406531930180. ✗ If `zip_dest="myarchive.zip"` is replaced by `read zip_dest` then output file name is obtained from the second line of standard input

Sub-Section Number :	6
Sub-Section Id :	64065382642
Question Shuffling Allowed :	Yes
Is Section Default? :	null

Question Number : 248 Question Id : 640653578000 Question Type : SA Calculator : None

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

Correct Marks : 7

Question Label : Short Answer Question


```
CUT(1) User
Commands
CUT(1)

NAME
    cut - remove sections from each line of files

SYNOPSIS
    cut OPTION... [FILE]...

DESCRIPTION
    Print selected parts of lines from each FILE to
    standard output.

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

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

    -b, --bytes=LIST
        select only these bytes

    -c, --characters=LIST
        select only these characters

    -d, --delimiter=DELIM
        use DELIM instead of TAB for field delimiter

    -f, --fields=LIST
        select only these fields; also print
        any line that contains no delimiter character, unless the
        -s
        option is specified
```

man sort command partial output

SORT(1)

User

Commands

SORT(1)

NAME

`sort` - `sort` lines of text files

SYNOPSIS

`sort` [OPTION]... [FILE]...

`sort` [OPTION]... --files0-from=F

DESCRIPTION

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:

-b, --ignore-leading-blanks
ignore leading blanks

-d, --dictionary-order
consider only blanks and alphanumeric

characters

-f, --ignore-case
fold lower case to upper case characters

-g, --general-numeric-sort
compare according to general numerical value

-i, --ignore-nonprinting
consider only printable characters

-M, --month-sort
compare (unknown) < 'JAN' < ... < 'DEC'

-h, --human-numeric-sort
compare human readable numbers (e.g., 2K 1G)

-n, --numeric-sort
compare according to string numerical value

-R, --random-sort
shuffle, but group identical keys. See
shuf(1)

--random-source=FILE
get random bytes from FILE

-r, --reverse
reverse the result of comparisons

--sort=WORD
sort according to WORD: general-numeric -g,
human-numeric -h, month -M, numeric -n, random -R, version
-V

-V, --version-sort
natural sort of (version) numbers within
text

The following are the contents of a `passwd` file. What will be the first line of the output of the command?

```
cat /etc/passwd|cut -d: -f3|sort -rn
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

65534

Question Number : 249 **Question Id :** 640653578003 **Question Type :** SA **Calculator :** None

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

Correct Marks : 7

Question Label : Short Answer Question

How many background processes are running after the end of execution of the following script?

```
for i in {1..11}; do
    sleep 10 &
done

for i in {1..11..2}; do
    kill % # kill the last background process created
done
```

Response Type : Numeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Text Areas : PlainText

Possible Answers :

5

Sub-Section Number : 7

Sub-Section Id : 64065382643

Question Shuffling Allowed : Yes

Is Section Default? : null

Question Number : 250 **Question Id :** 640653578006 **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 the output of the following script?

```
read -n 4 line < <(echo abcdef)
echo $line$line
```


Hint:

```
$ cat < <(echo 123456) # send the stdout from echo command  
to cat as stdin  
123456
```

```
$ help read
```

```
read: read [-ers] [-a array] [-d delim] [-i text] [-n  
nchars] [-N nchars] [-p prompt] [-t timeout] [-u fd] [name  
...]
```

Read a line from the standard input and split it into fields.

Reads a single line from the standard input, or from file descriptor FD

if the -u option is supplied. The line is split into fields as with word

splitting, and the first word is assigned to the first NAME, the second

word to the second NAME, and so on, with any leftover words assigned to

the last NAME. Only the characters found in \$IFS are recognized as word

delimiters.

If no NAMES are supplied, the line read is stored in the REPLY variable.

Options:

- a array assign the words read to sequential indices of the array
 variable ARRAY, starting at zero
- d delim continue until the first character of DELIM is read, rather
 than newline
- e use Readline to obtain the line
- i text use TEXT as the initial text for Readline
- n nchars return after reading NCHARS characters rather than waiting
 for a newline, but honor a delimiter if fewer than
 NCHARS characters are read before the delimiter
- N nchars return only after reading exactly NCHARS characters, unless
 EOF is encountered or read times out, ignoring any
 delimiter
- p prompt output the string PROMPT without a trailing newline before
 attempting to read
- r do not allow backslashes to escape any characters
- s do not echo input coming from a terminal
- t timeout time out and return failure if a

complete line of
input is not read within TIMEOUT seconds.
The value of the
TMOUT variable is the default timeout.
TIMEOUT may be a
fractional number. If TIMEOUT is 0, read
returns
immediately, without trying to read any
data, returning
success only if input is available on the
specified
file descriptor. The exit status is
greater than 128
if the timeout is exceeded
-u fd read from file descriptor FD instead of
the standard input

Exit Status:

The return code is zero, unless end-of-file is
encountered, read times out
(in which case it's greater than 128), a variable
assignment error occurs,
or an invalid file descriptor is supplied as the
argument to -u.

Response Type : Alphanumeric

Evaluation Required For SA : Yes

Show Word Count : Yes

Answers Type : Equal

Answers Case Sensitive : Yes

Text Areas : PlainText

Possible Answers :

abcdabcd