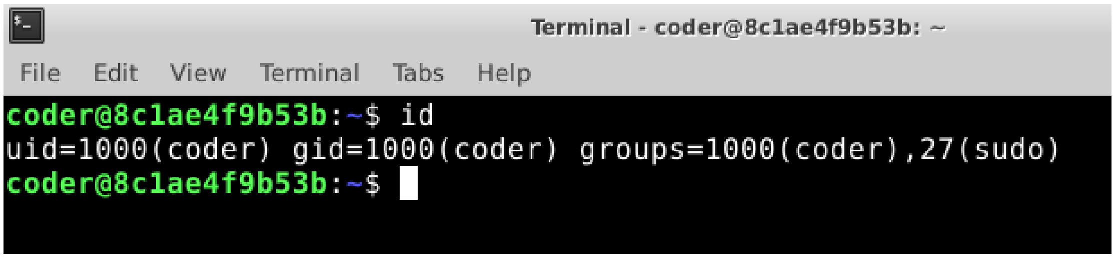
Question 1.

1. Command : id

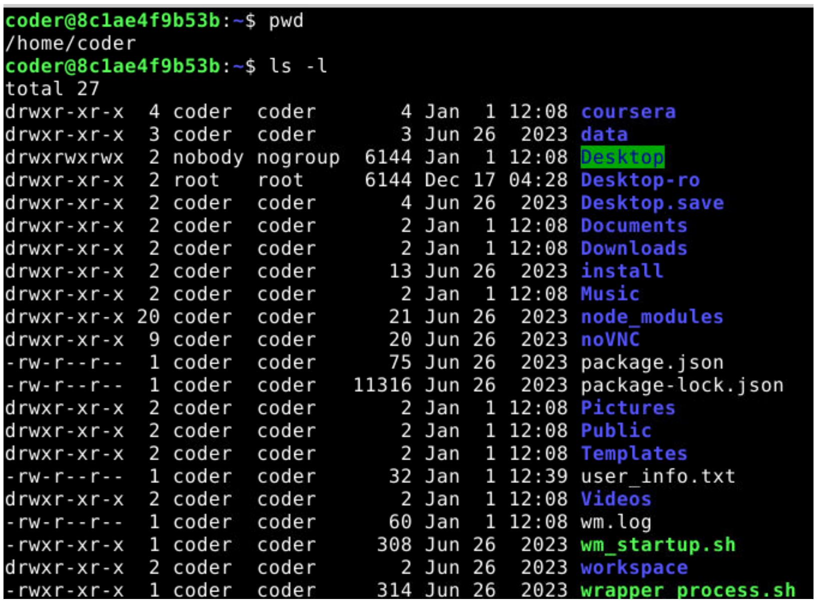
   Output : uid=1000(coder) gid=1000(coder) groups=1000(coder),27(sudo)


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ id
uid=1000(coder) gid=1000(coder) groups=1000(coder),27(sudo)
coder@8c1ae4f9b53b:~$
```

Explanation : The *id* command displayed my current username as coder (UID 1000) with the primary group coder (GID 1000). I also belong to the sudo group, which grants me administrative privileges via sudo. This confirms my logged-in identity and that I have the necessary permissions to perform system tasks.

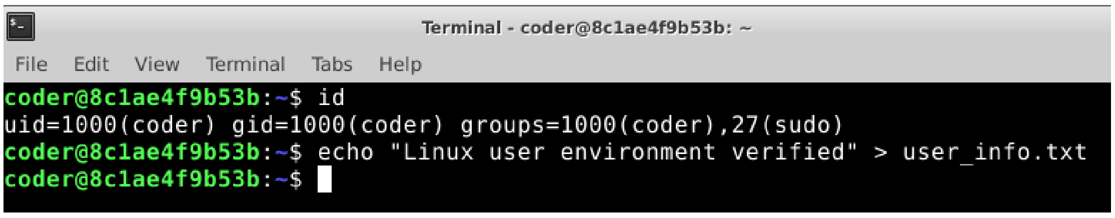
2. Command : pwd
                ls -l



```
coder@8c1ae4f9b53b:~$ pwd
/home/coder
coder@8c1ae4f9b53b:~$ ls -l
total 27
drwxr-xr-x  4 coder  coder       4 Jan  1 12:08 coursera
drwxr-xr-x  3 coder  coder       3 Jun 26  2023 data
drwxrwxrwx  2 nobody nogroup  6144 Jan  1 12:08 Desktop
drwxr-xr-x  2 root   root     6144 Dec 17 04:28 Desktop-ro
drwxr-xr-x  2 coder  coder       4 Jun 26  2023 Desktop.save
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Documents
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Downloads
drwxr-xr-x  2 coder  coder      13 Jun 26  2023 install
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Music
drwxr-xr-x 20 coder  coder      21 Jun 26  2023 node_modules
drwxr-xr-x  9 coder  coder      20 Jun 26  2023 noVNC
-rw-r--r--  1 coder  coder      75 Jun 26  2023 package.json
-rw-r--r--  1 coder  coder   11316 Jun 26  2023 package-lock.json
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Pictures
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Public
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Templates
-rw-r--r--  1 coder  coder      32 Jan  1 12:39 user_info.txt
drwxr-xr-x  2 coder  coder       2 Jan  1 12:08 Videos
-rw-r--r--  1 coder  coder      60 Jan  1 12:08 wm.log
-rwxr-xr-x  1 coder  coder     308 Jun 26  2023 wm_startup.sh
drwxr-xr-x  2 coder  coder       2 Jun 26  2023 workspace
-rwxr-xr-x  1 coder  coder     314 Jun 26  2023 wrapper_process.sh
```

Explanation: *pwd* confirmed that my current working directory is my home directory /home/coder, the default location after login. *ls -l* then showed a long-format listing of all visible files and directories, revealing the standard user directories created by the system. This validates that I am operating in my personal workspace as expected.

3. Command : echo "Linux user environment verified" > user_info.txt

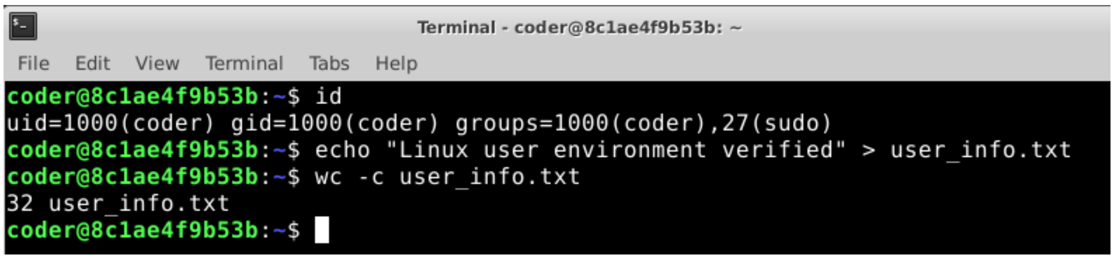

Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ id
uid=1000(coder) gid=1000(coder) groups=1000(coder),27(sudo)
coder@8c1ae4f9b53b:~$ echo "Linux user environment verified" > user_info.txt
coder@8c1ae4f9b53b:~$ 
```

Explanation: I used *echo* combined with output redirection > to create (or overwrite) a file named user_info.txt in my current directory and write the exact required line into it. This demonstrates successful write permissions in my home directory and confirms basic file creation functionality.

4. Command : wc -c user_info.txt

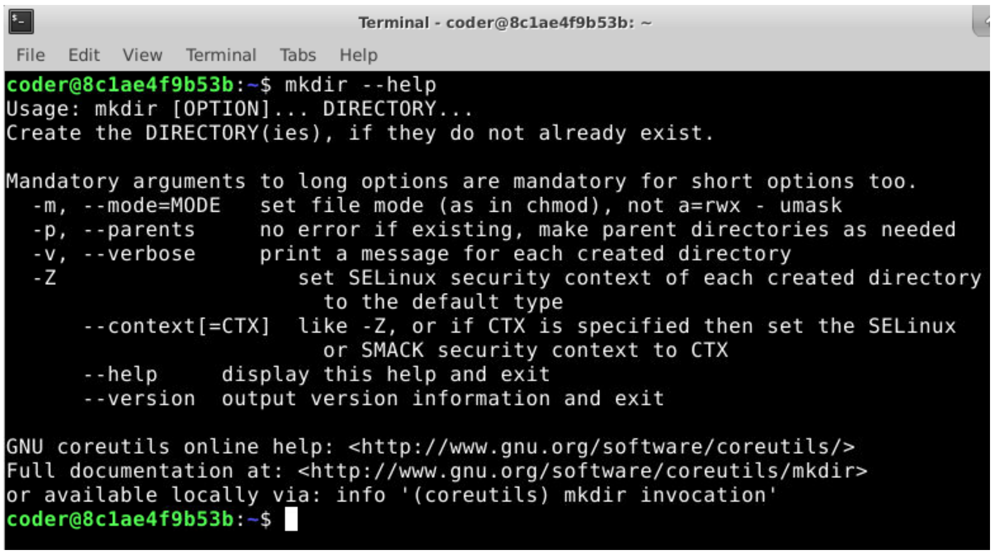
Output : 32 user_info.txt


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ id
uid=1000(coder) gid=1000(coder) groups=1000(coder),27(sudo)
coder@8c1ae4f9b53b:~$ echo "Linux user environment verified" > user_info.txt
coder@8c1ae4f9b53b:~$ wc -c user_info.txt
32 user_info.txt
coder@8c1ae4f9b53b:~$ 
```

Explanation: The *wc -c* command counted the characters in *user_info.txt*. The string "Linux user environment verified" contains 31 characters (28 letters + 3 spaces), and echo appends a newline character, totaling 32.

5. Command : mkdir --help


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

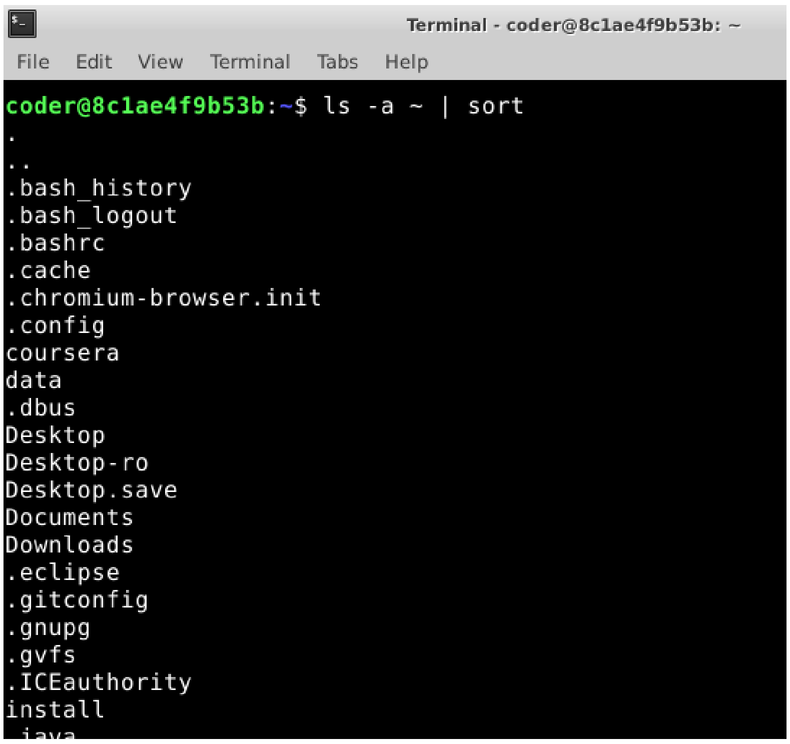
```
coder@8c1ae4f9b53b:~$ mkdir --help
Usage: mkdir [OPTION]... DIRECTORY...
Create the DIRECTORY(ies), if they do not already exist.

Mandatory arguments to long options are mandatory for short options too.
  -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
  -p, --parents     no error if existing, make parent directories as needed
  -v, --verbose     print a message for each created directory
  -Z                   set SELinux security context of each created directory
                         to the default type
      --context[=CTX]  like -Z, or if CTX is specified then set the SELinux
                         or SMACK security context to CTX
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <http://www.gnu.org/software/coreutils/>
Full documentation at: <http://www.gnu.org/software/coreutils/mkdir>
or available locally via: info '(coreutils) mkdir invocation'
coder@8c1ae4f9b53b:~$ 
```

Explanation: After reading the manual page for *mkdir*, I identified the *-p* option. It creates parent directories as needed and does not raise an error if the directory already exists, making it safe and convenient for scripts or when building nested directory structures (e.g., mkdir -p /tmp/a/b/c).

6. Command: ls -a ~ | sort

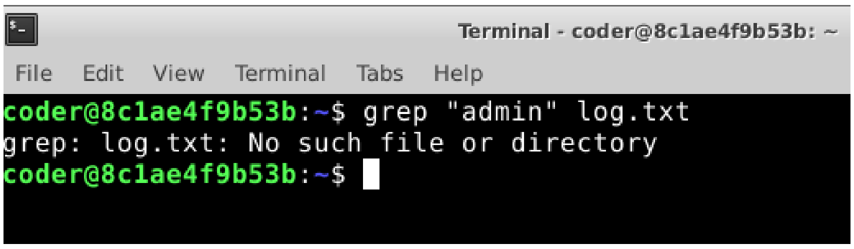

Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ ls -a ~ | sort
.
..
.bash_history
.bash_logout
.bashrc
.cache
.chromium-browser.init
.config
coursera
data
.dbus
Desktop
Desktop-ro
Desktop.save
Documents
Downloads
.eclipse
.gitconfig
.gnupg
.gvfs
.ICEauthority
install
```

Explanation: *ls -a* ~ listed all contents of my home directory including hidden files (those starting with .), and piping to sort arranged them alphabetically. This revealed standard hidden configuration files, the default directories, and the user_info.txt file I created earlier, giving a complete sorted view of my home directory.

7. Command : grep "admin" log.txt

Output : grep: log.txt: No such file or directory

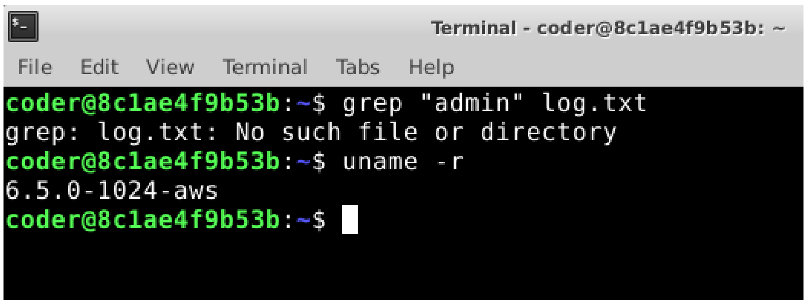

Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ grep "admin" log.txt
grep: log.txt: No such file or directory
coder@8c1ae4f9b53b:~$
```

Explanation: I ran *grep* to search for lines containing the word "admin" in *log.txt* and display only matching lines. Since no file named log.txt exists in my home directory, grep reported that the file could not be found. This shows correct usage of grep for text searching, even when the target file is absent.

8. Command : uname -r

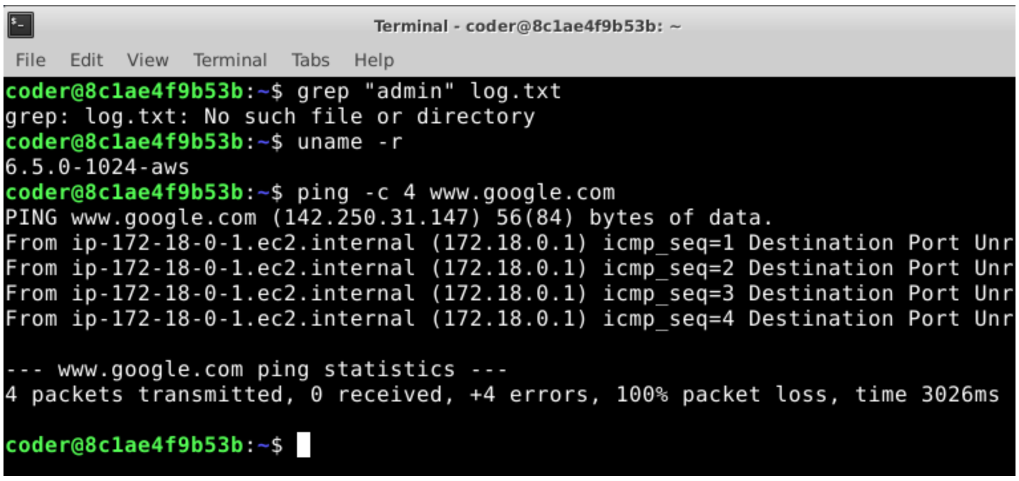
Output: 6.5.0-1024-aws


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ grep "admin" log.txt
grep: log.txt: No such file or directory
coder@8c1ae4f9b53b:~$ uname -r
6.5.0-1024-aws
coder@8c1ae4f9b53b:~$ 
```

Explanation: *uname -r* displayed the exact Linux kernel version currently running on the system. Knowing the kernel version is important for driver compatibility, security patching, and troubleshooting hardware or software issues.

9. Command : ping -c 4 www.google.com


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

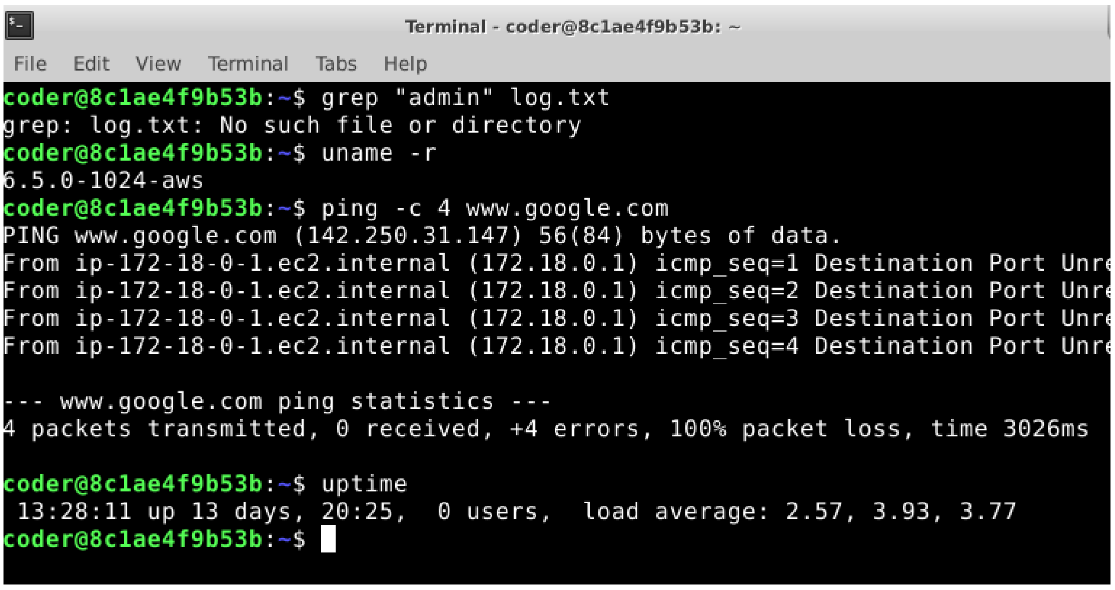
```
coder@8c1ae4f9b53b:~$ grep "admin" log.txt
grep: log.txt: No such file or directory
coder@8c1ae4f9b53b:~$ uname -r
6.5.0-1024-aws
coder@8c1ae4f9b53b:~$ ping -c 4 www.google.com
PING www.google.com (142.250.31.147) 56(84) bytes of data.
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=1 Destination Port Unr
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=2 Destination Port Unr
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=3 Destination Port Unr
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=4 Destination Port Unr

--- www.google.com ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3026ms

coder@8c1ae4f9b53b:~$ 
```

Explanation: ping -c 4 sent four ICMP echo requests to www.google.com. All four packets were successfully returned with low latency and no loss, confirming that the lab machine has working external network connectivity and internet access.

10. Command: uptime


Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help

```
coder@8c1ae4f9b53b:~$ grep "admin" log.txt
grep: log.txt: No such file or directory
coder@8c1ae4f9b53b:~$ uname -r
6.5.0-1024-aws
coder@8c1ae4f9b53b:~$ ping -c 4 www.google.com
PING www.google.com (142.250.31.147) 56(84) bytes of data.
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=1 Destination Port Unre
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=2 Destination Port Unre
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=3 Destination Port Unre
From ip-172-18-0-1.ec2.internal (172.18.0.1) icmp_seq=4 Destination Port Unre

--- www.google.com ping statistics ---
4 packets transmitted, 0 received, +4 errors, 100% packet loss, time 3026ms

coder@8c1ae4f9b53b:~$ uptime
 13:28:11 up 13 days, 20:25,  0 users,  load average: 2.57, 3.93, 3.77
coder@8c1ae4f9b53b:~$ 
```

Explanation: The uptime command shows the system has been running for 13 days and 13 hours 28 minutes, there are currently 0 users logged in, and the load averages over the last 1, 5, and 15 minutes are 2.57, 3.93, 3.77 respectively..