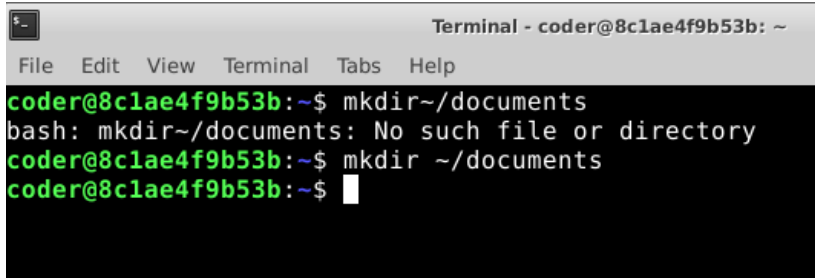


Question 2.

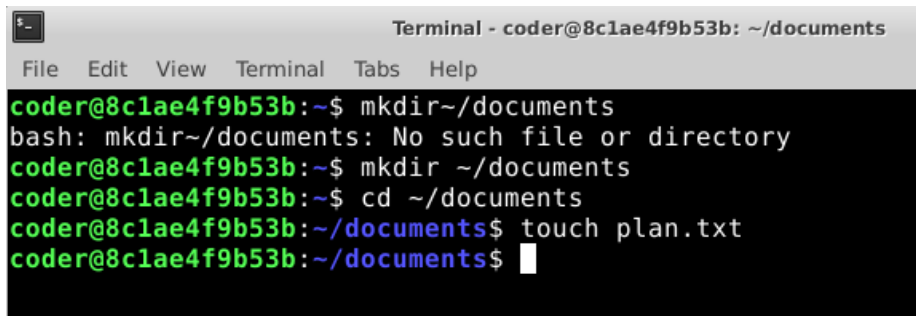
1. Command: `mkdir ~/documents`



```
Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$
```

Explanation: *mkdir* creates a new directory. `~` is shorthand for your home directory (`/home/coder`). This gives you a clean folder called `documents` to store all project files.

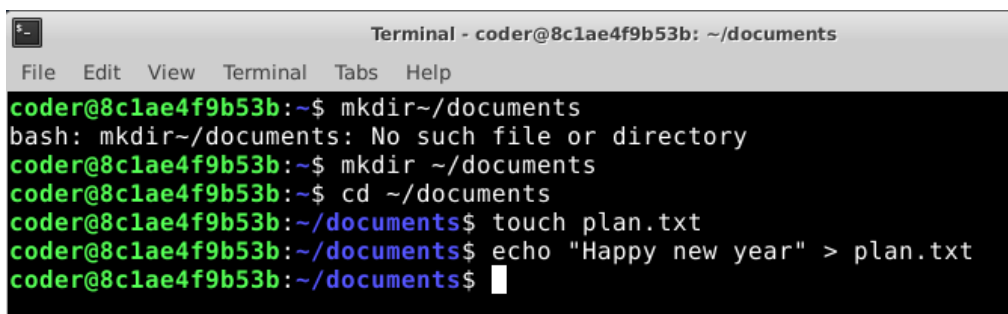
2. Command: `cd ~/documents, touch plan.txt`



```
Terminal - coder@8c1ae4f9b53b: ~/documents
File Edit View Terminal Tabs Help
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$ cd ~/documents
coder@8c1ae4f9b53b:~/documents$ touch plan.txt
coder@8c1ae4f9b53b:~/documents$
```

Explanation: *cd* changes your current location to the new `documents` folder. *touch plan.txt* creates an empty file named `plan.txt`.

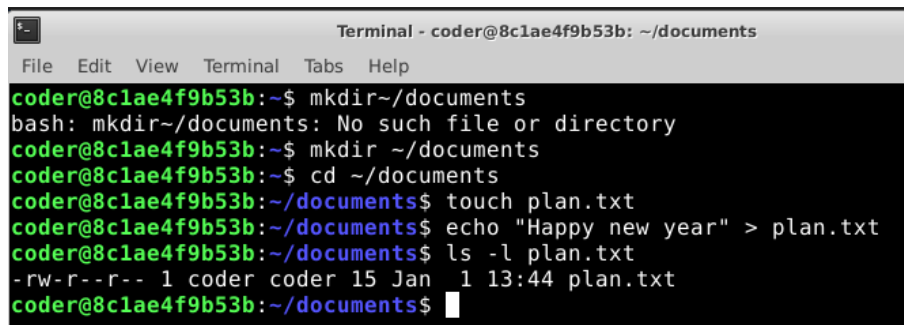
3. Command: `echo "Happy new year" > plan.txt`



```
Terminal - coder@8c1ae4f9b53b: ~/documents
File Edit View Terminal Tabs Help
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$ cd ~/documents
coder@8c1ae4f9b53b:~/documents$ touch plan.txt
coder@8c1ae4f9b53b:~/documents$ echo "Happy new year" > plan.txt
coder@8c1ae4f9b53b:~/documents$
```

Explanation: *echo* prints the text, and *>* redirects it into *plan.txt*, overwriting the file with your chosen note. You can write any short reminder you like.

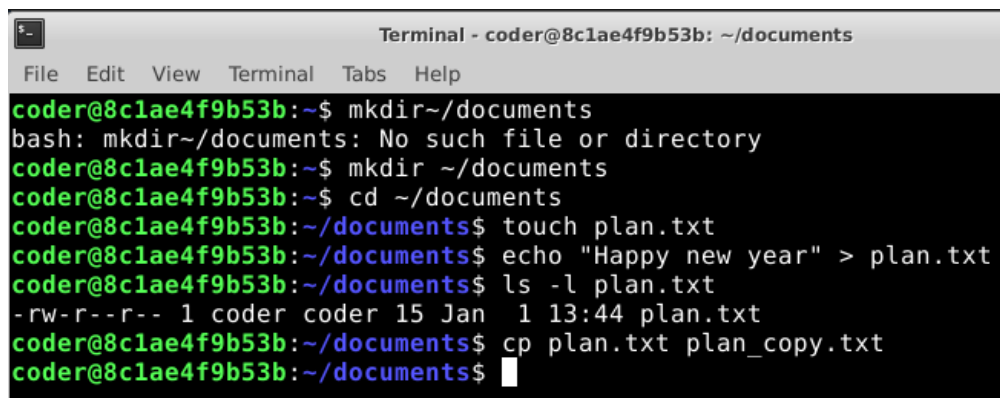
4. Command: `ls -l plan.txt`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~/documents" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$ cd ~/documents
coder@8c1ae4f9b53b:~/documents$ touch plan.txt
coder@8c1ae4f9b53b:~/documents$ echo "Happy new year" > plan.txt
coder@8c1ae4f9b53b:~/documents$ ls -l plan.txt
-rw-r--r-- 1 coder coder 15 Jan  1 13:44 plan.txt
coder@8c1ae4f9b53b:~/documents$
```

Explanation: *ls -l* shows detailed (long) info: permissions (*-rw-rw-r--*), link count (*1*), owner (*coder*), group (*coder*), date/time, and filename. Your username *coder* must appear here as owner.

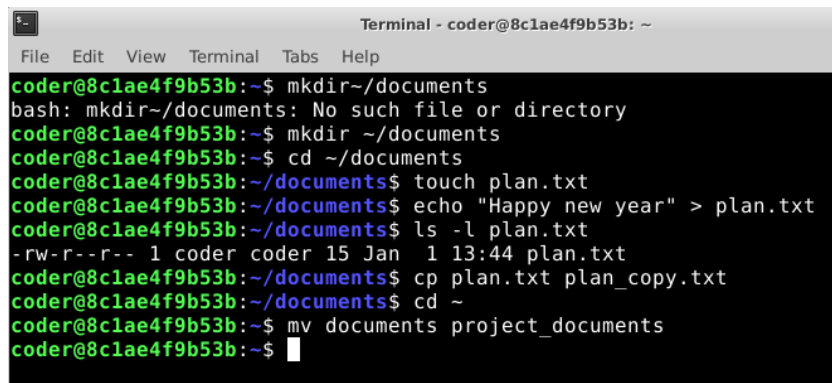
5. Command: `cp plan.txt plan_copy.txt`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~/documents" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$ cd ~/documents
coder@8c1ae4f9b53b:~/documents$ touch plan.txt
coder@8c1ae4f9b53b:~/documents$ echo "Happy new year" > plan.txt
coder@8c1ae4f9b53b:~/documents$ ls -l plan.txt
-rw-r--r-- 1 coder coder 15 Jan  1 13:44 plan.txt
coder@8c1ae4f9b53b:~/documents$ cp plan.txt plan_copy.txt
coder@8c1ae4f9b53b:~/documents$
```

Explanation: *cp* (copy) creates an identical duplicate named *plan_copy.txt* in the same directory. Both files now have the same content.

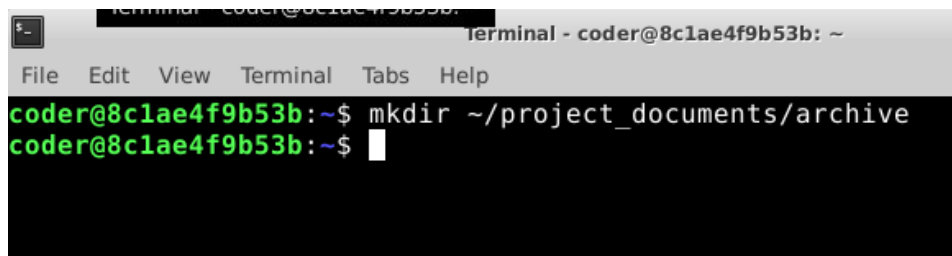
6. Command: `cd ~, mv documents project_documents`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
coder@8c1ae4f9b53b:~$ mkdir~/documents
bash: mkdir~/documents: No such file or directory
coder@8c1ae4f9b53b:~$ mkdir ~/documents
coder@8c1ae4f9b53b:~$ cd ~/documents
coder@8c1ae4f9b53b:~/documents$ touch plan.txt
coder@8c1ae4f9b53b:~/documents$ echo "Happy new year" > plan.txt
coder@8c1ae4f9b53b:~/documents$ ls -l plan.txt
-rw-r--r-- 1 coder coder 15 Jan  1 13:44 plan.txt
coder@8c1ae4f9b53b:~/documents$ cp plan.txt plan_copy.txt
coder@8c1ae4f9b53b:~/documents$ cd ~
coder@8c1ae4f9b53b:~$ mv documents project_documents
coder@8c1ae4f9b53b:~$
```

Explanation: *mv* (move) is also used for renaming. This renames the folder from *documents* to *project_documents*. All files inside (*plan.txt* and *plan_copy.txt*) automatically move with it.

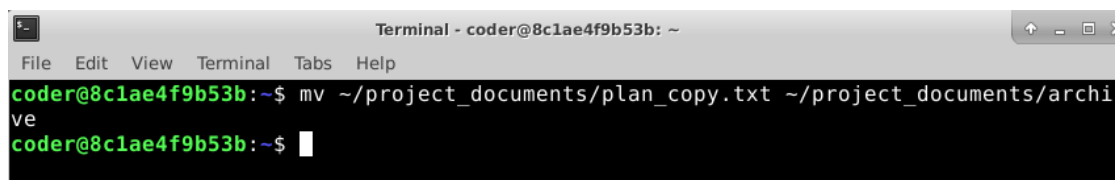
7. Command: `mkdir ~/project_documents/archive`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The terminal shows the following commands and output:

```
coder@8c1ae4f9b53b:~$ mkdir ~/project_documents/archive
coder@8c1ae4f9b53b:~$
```

Explanation: Creates a new subfolder called *archive* inside *project_documents* – perfect for storing old or backup files.

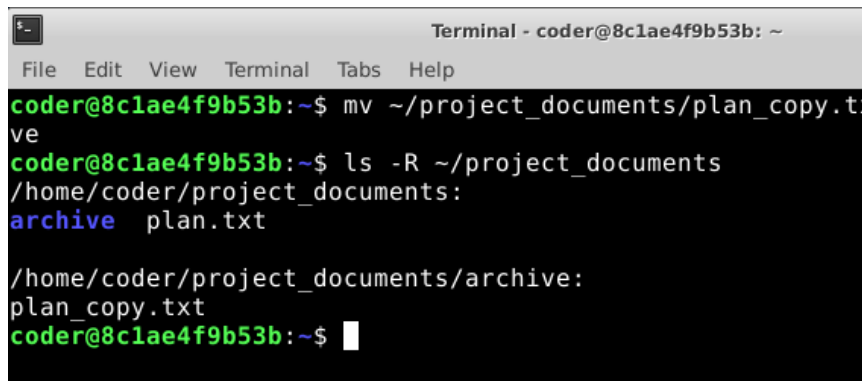
8. Command: `mv ~/project_documents/plan_copy.txt ~/project_documents/archive/`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help) and window controls. The terminal shows the following commands and output:

```
coder@8c1ae4f9b53b:~$ mv ~/project_documents/plan_copy.txt ~/project_documents/archive/
coder@8c1ae4f9b53b:~$
```

Explanation: *mv* moves *plan_copy.txt* from the main project folder into the *archive* subfolder, keeping things organized.

9. Command: `ls -R ~/project_documents`

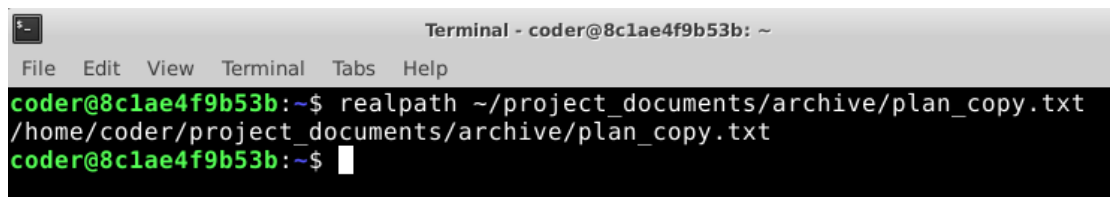
A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is "coder@8c1ae4f9b53b:~\$". The first command is "mv ~/project_documents/plan_copy.txt", followed by "ve" on the next line. The second command is "ls -R ~/project_documents". The output shows the directory structure: "/home/coder/project_documents:" followed by "archive" and "plan.txt" on the next line. Then, it shows the contents of the archive directory: "/home/coder/project_documents/archive:" followed by "plan_copy.txt" on the next line. The prompt returns to "coder@8c1ae4f9b53b:~\$".

```
Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help
coder@8c1ae4f9b53b:~$ mv ~/project_documents/plan_copy.txt
ve
coder@8c1ae4f9b53b:~$ ls -R ~/project_documents
/home/coder/project_documents:
archive  plan.txt

/home/coder/project_documents/archive:
plan_copy.txt
coder@8c1ae4f9b53b:~$
```

Explanation: `ls -R` means “recursive”. It lists the contents of *project_documents* and then drills down into every subdirectory (here, *archive*), showing the complete tree structure.

10. Command: `realpath ~/project_documents/archive/plan_copy.txt`

A terminal window titled "Terminal - coder@8c1ae4f9b53b: ~" with a menu bar (File, Edit, View, Terminal, Tabs, Help). The prompt is "coder@8c1ae4f9b53b:~\$". The command is "realpath ~/project_documents/archive/plan_copy.txt". The output is the absolute path: "/home/coder/project_documents/archive/plan_copy.txt". The prompt returns to "coder@8c1ae4f9b53b:~\$".

```
Terminal - coder@8c1ae4f9b53b: ~
File Edit View Terminal Tabs Help
coder@8c1ae4f9b53b:~$ realpath ~/project_documents/archive/plan_copy.txt
/home/coder/project_documents/archive/plan_copy.txt
coder@8c1ae4f9b53b:~$
```

Explanation: This prints the full absolute path of the moved file, confirming it is now located inside the archive folder.

These commands demonstrate core Linux file management skills: creating, copying, moving, renaming, and inspecting files/directories — all safely within your own user space.

