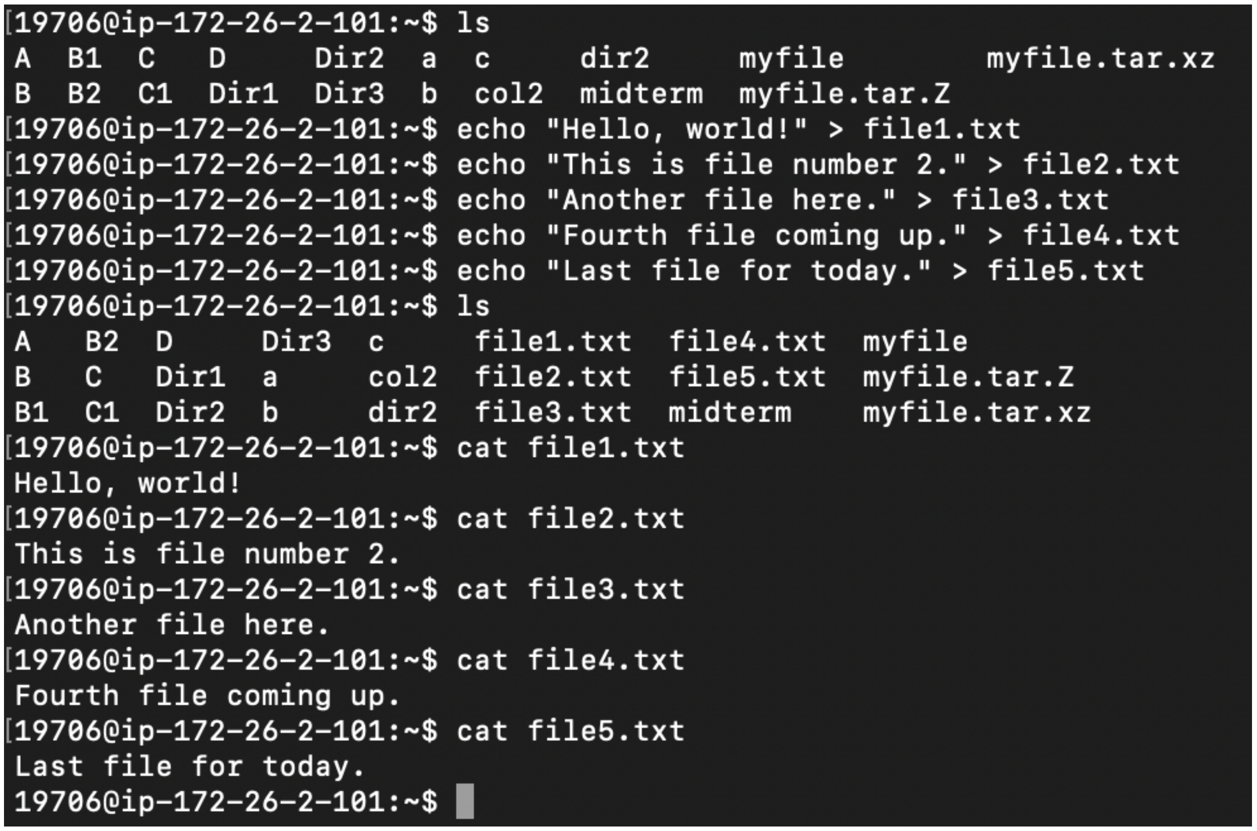
Question no.1

args v exec

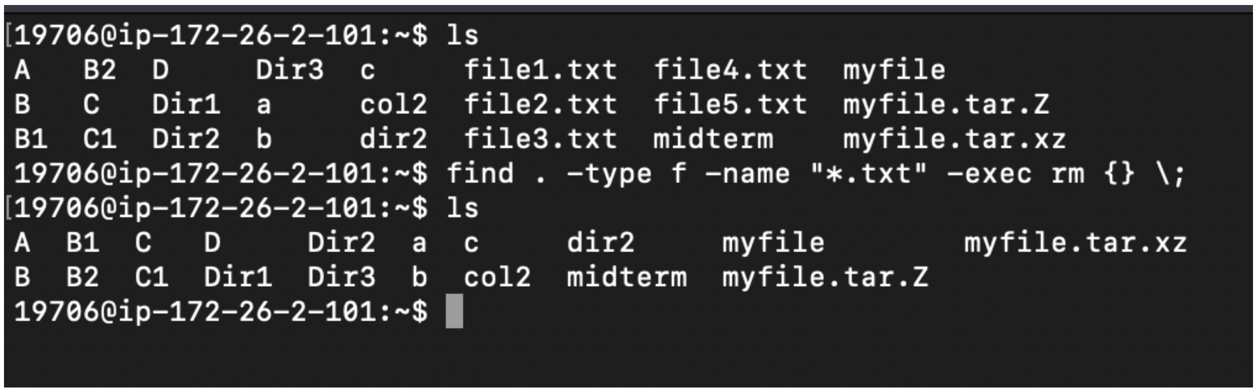
- Setup: You need to create at least 5 text files with extensions .txt. Each file can just be 1 line. You can use echo or any text editor to create these files.
  ⇨ the commands and outputs to create 5 text files with the ".txt" extension:



```
[19706@ip-172-26-2-101:~$ ls
A  B1  C   D     Dir2  a  c     dir2     myfile        myfile.tar.xz
B  B2  C1  Dir1  Dir3  b  col2  midterm  myfile.tar.Z
[19706@ip-172-26-2-101:~$ echo "Hello, world!" > file1.txt
[19706@ip-172-26-2-101:~$ echo "This is file number 2." > file2.txt
[19706@ip-172-26-2-101:~$ echo "Another file here." > file3.txt
[19706@ip-172-26-2-101:~$ echo "Fourth file coming up." > file4.txt
[19706@ip-172-26-2-101:~$ echo "Last file for today." > file5.txt
[19706@ip-172-26-2-101:~$ ls
A   B2  D     Dir3  c     file1.txt  file4.txt  myfile
B   C   Dir1  a     col2  file2.txt  file5.txt  myfile.tar.Z
B1  C1  Dir2  b     dir2  file3.txt  midterm    myfile.tar.xz
[19706@ip-172-26-2-101:~$ cat file1.txt
Hello, world!
[19706@ip-172-26-2-101:~$ cat file2.txt
This is file number 2.
[19706@ip-172-26-2-101:~$ cat file3.txt
Another file here.
[19706@ip-172-26-2-101:~$ cat file4.txt
Fourth file coming up.
[19706@ip-172-26-2-101:~$ cat file5.txt
Last file for today.
19706@ip-172-26-2-101:~$
```

- The find command supports the -exec option that allows arbitrary commands to be performed on found files. The following are equivalent.
- find ./ -type f -name "*.txt" -exec rm {} \;

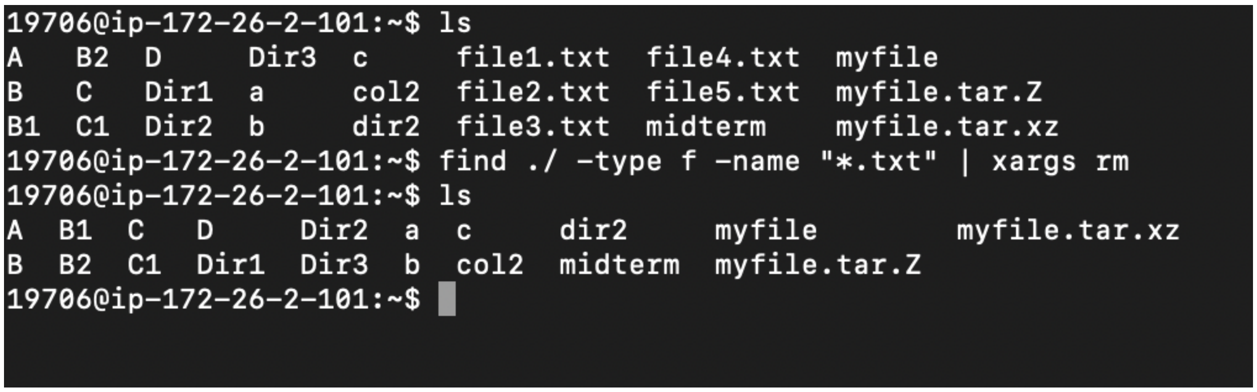
  ⇨ a



```
19706@ip-172-26-2-101:~$ ls
A    B2   D      Dir3   c      file1.txt  file4.txt  myfile
B    C    Dir1   a      col2   file2.txt  file5.txt  myfile.tar.Z
B1   C1   Dir2   b      dir2   file3.txt  midterm    myfile.tar.xz
19706@ip-172-26-2-101:~$ find . -type f -name "*.txt" -exec rm {} \;
19706@ip-172-26-2-101:~$ ls
A   B1   C    D      Dir2   a   c      dir2     myfile         myfile.tar.xz
B   B2   C1   Dir1   Dir3   b   col2   midterm  myfile.tar.Z
19706@ip-172-26-2-101:~$ 
```

- find ./ -type f -name "*.txt" | xargs rm

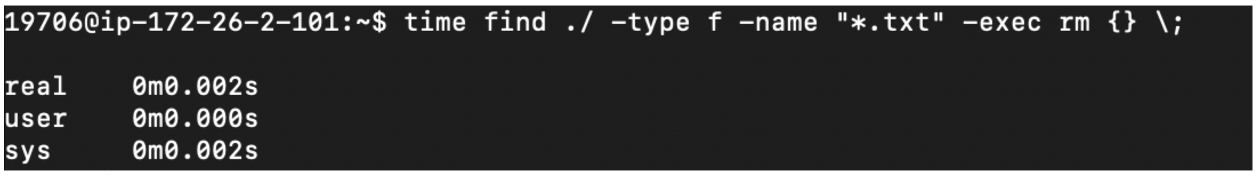
  ⇨ s



```
19706@ip-172-26-2-101:~$ ls
A    B2   D      Dir3   c      file1.txt  file4.txt  myfile
B    C    Dir1   a      col2   file2.txt  file5.txt  myfile.tar.Z
B1   C1   Dir2   b      dir2   file3.txt  midterm    myfile.tar.xz
19706@ip-172-26-2-101:~$ find ./ -type f -name "*.txt" | xargs rm
19706@ip-172-26-2-101:~$ ls
A   B1   C    D      Dir2   a   c      dir2     myfile         myfile.tar.xz
B   B2   C1   Dir1   Dir3   b   col2   midterm  myfile.tar.Z
19706@ip-172-26-2-101:~$ 
```

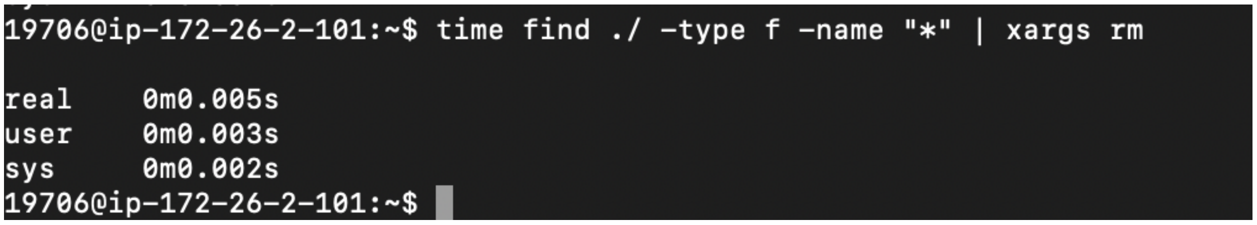
- use the time command to time each one to see if which command above is faster? Give me the output from the time command for each

  ⇨ The time outputs indicate that the first command completed faster with a real-time of 0.002s, while the second command took a little longer with a real-time of 0.005s. However, the difference in speed is very small and may not be noticeable for small numbers of files. For larger numbers of files, the difference in speed may be more significant.



```
19706@ip-172-26-2-101:~$ time find ./ -type f -name "*.txt" -exec rm {} \;

real	0m0.002s
user	0m0.000s
sys	0m0.002s
```



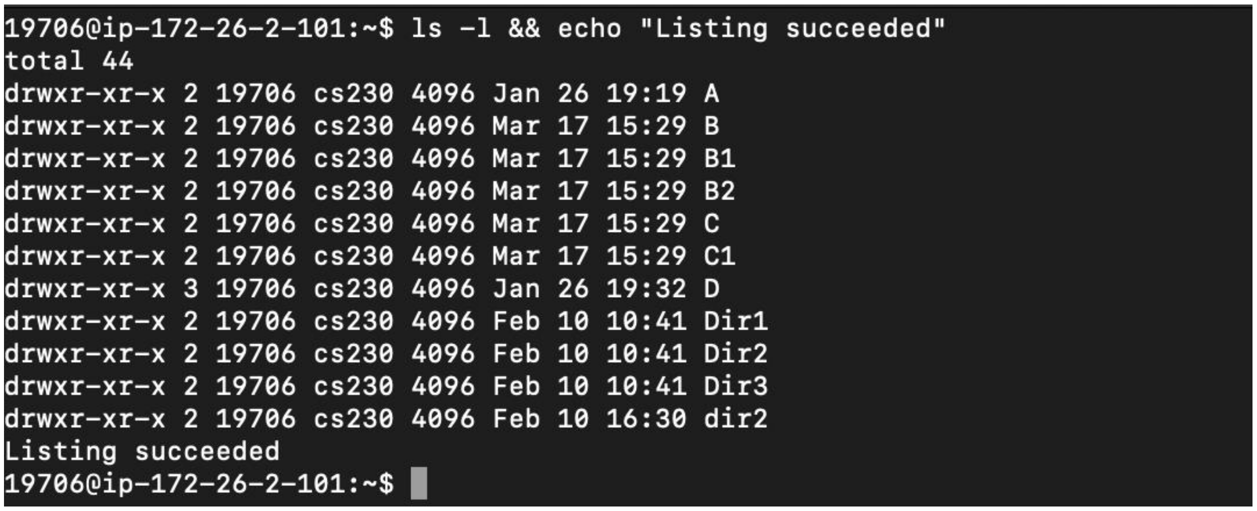
```
19706@ip-172-26-2-101:~$ time find ./ -type f -name "*" | xargs rm

real	0m0.005s
user	0m0.003s
sys	0m0.002s
19706@ip-172-26-2-101:~$
```

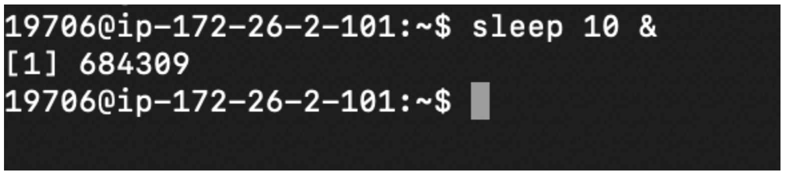
Question no.2
special characters
Linux commands use many special characters.

- && is different from &

⇨ && operator is used to chain commands together so that the second command is only executed if the first command succeeds.


```
19706@ip-172-26-2-101:~$ ls -l && echo "Listing succeeded"
total 44
drwxr-xr-x 2 19706 cs230 4096 Jan 26 19:19 A
drwxr-xr-x 2 19706 cs230 4096 Mar 17 15:29 B
drwxr-xr-x 2 19706 cs230 4096 Mar 17 15:29 B1
drwxr-xr-x 2 19706 cs230 4096 Mar 17 15:29 B2
drwxr-xr-x 2 19706 cs230 4096 Mar 17 15:29 C
drwxr-xr-x 2 19706 cs230 4096 Mar 17 15:29 C1
drwxr-xr-x 3 19706 cs230 4096 Jan 26 19:32 D
drwxr-xr-x 2 19706 cs230 4096 Feb 10 10:41 Dir1
drwxr-xr-x 2 19706 cs230 4096 Feb 10 10:41 Dir2
drwxr-xr-x 2 19706 cs230 4096 Feb 10 10:41 Dir3
drwxr-xr-x 2 19706 cs230 4096 Feb 10 16:30 dir2
Listing succeeded
19706@ip-172-26-2-101:~$
```

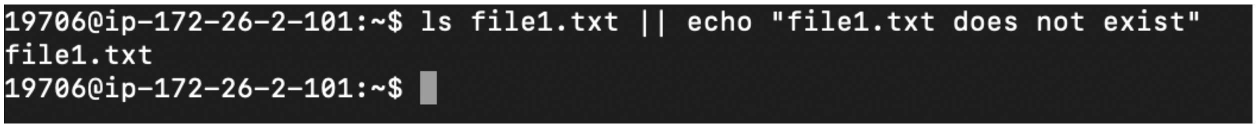
⇨ & operator is used to running a command in the background, allowing the user to continue using the terminal while the command runs.


```
19706@ip-172-26-2-101:~$ sleep 10 &
[1] 684309
19706@ip-172-26-2-101:~$
```

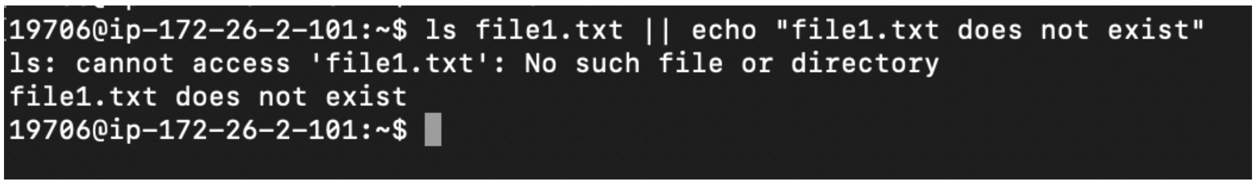
- || is different from |

⇨ the '||' operator is used for the conditional execution of a command. It executes the second command only if the first command fails.

⇨ In the first example, the first command executes it gets success, so the second command doesn't run.

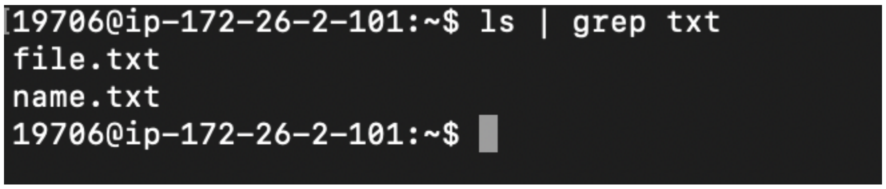

```
19706@ip-172-26-2-101:~$ ls file1.txt || echo "file1.txt does not exist"
file1.txt
19706@ip-172-26-2-101:~$
```

⇨ In the second example, the first command executes it get fails, so the second command run.



```
19706@ip-172-26-2-101:~$ ls file1.txt || echo "file1.txt does not exist"
ls: cannot access 'file1.txt': No such file or directory
file1.txt does not exist
19706@ip-172-26-2-101:~$
```

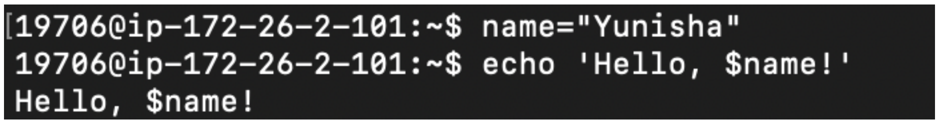
⇨ The | operator is used for piping the output of one command as input to another command.



```
19706@ip-172-26-2-101:~$ ls | grep txt
file.txt
name.txt
19706@ip-172-26-2-101:~$
```

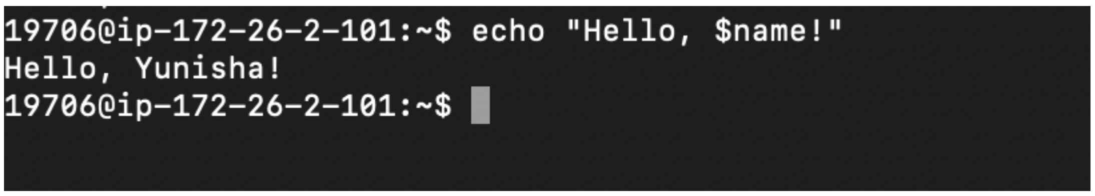
- single-quote (') is different from a double quote (")

⇨ Single quotes are used to denote a string literal that is not subject to any interpretation or expansion, including variables and commands.



```
19706@ip-172-26-2-101:~$ name="Yunisha"
19706@ip-172-26-2-101:~$ echo 'Hello, $name!'
Hello, $name!
```

⇨ Double quotes are used to allow variable expansion and command substitution, while single quotes treat everything literally without any special interpretation.



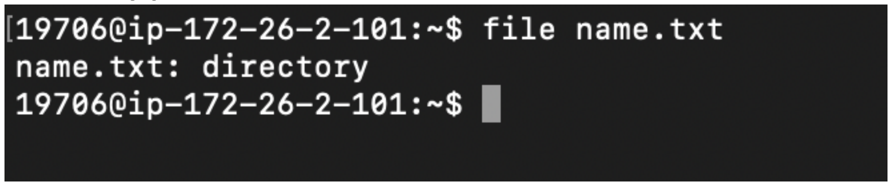
```
19706@ip-172-26-2-101:~$ echo "Hello, $name!"
Hello, Yunisha!
19706@ip-172-26-2-101:~$
```

Question no.3

Go through Lesson 7 from Webminal: Locate its file and its type Webminal_Lesson7.

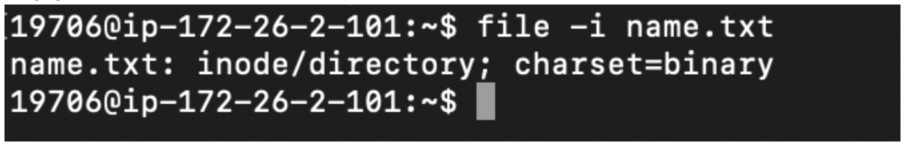
a) File:

i) Type of a file:



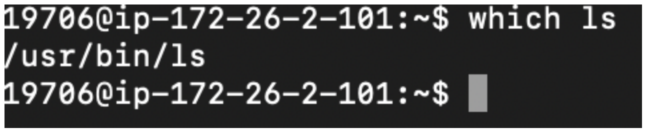
```
19706@ip-172-26-2-101:~$ file name.txt
name.txt: directory
19706@ip-172-26-2-101:~$
```

ii) Type of a file as ASCII text:



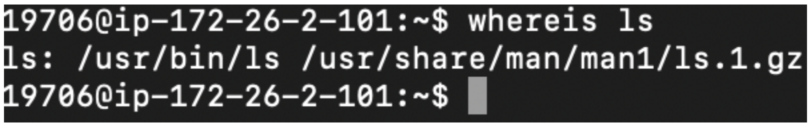
```
19706@ip-172-26-2-101:~$ file -i name.txt
name.txt: inode/directory; charset=binary
19706@ip-172-26-2-101:~$
```

b) Whereis:

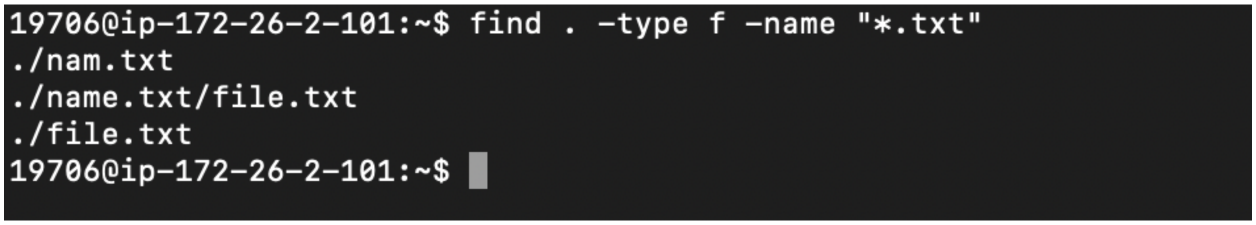


```
19706@ip-172-26-2-101:~$ which ls
/usr/bin/ls
19706@ip-172-26-2-101:~$
```

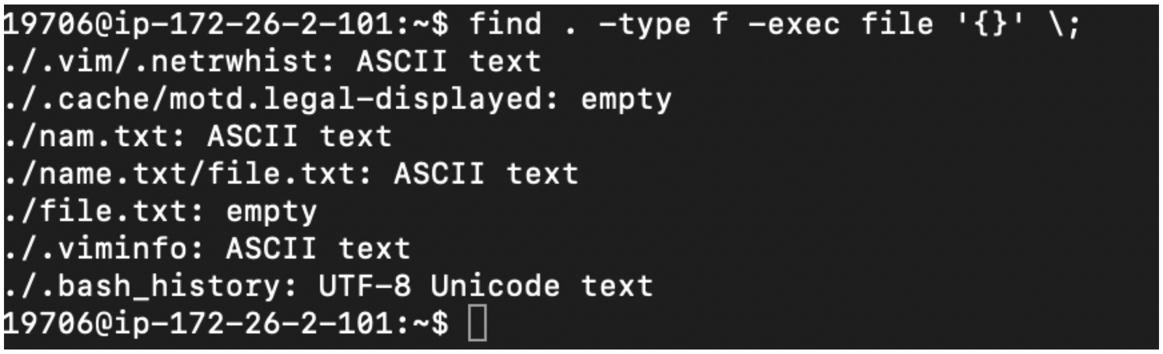
c) Which:


```
19706@ip-172-26-2-101:~$ whereis ls
ls: /usr/bin/ls /usr/share/man/man1/ls.1.gz
19706@ip-172-26-2-101:~$
```

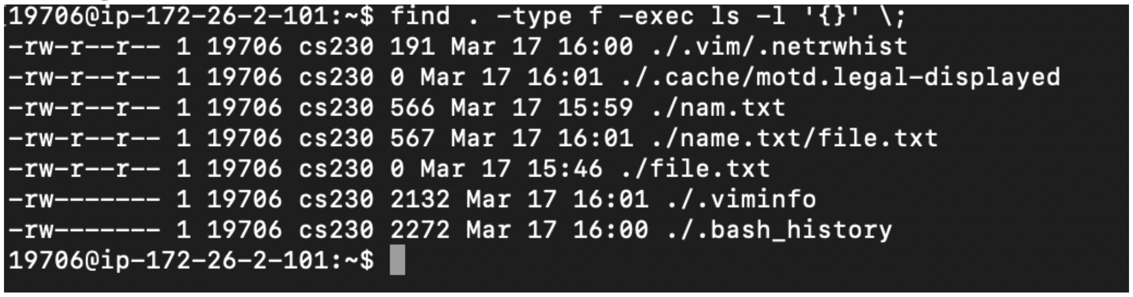
d) Find:


```
19706@ip-172-26-2-101:~$ find . -type f -name "*.txt"
./nam.txt
./name.txt/file.txt
./file.txt
19706@ip-172-26-2-101:~$
```

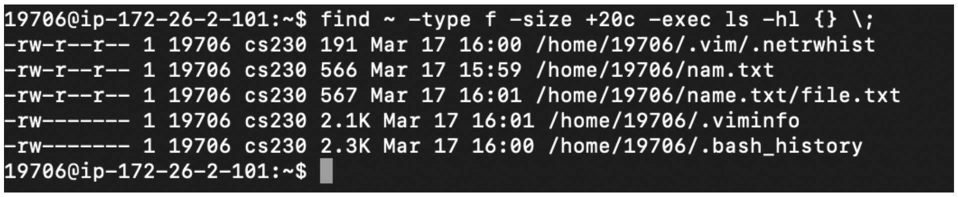
i) To find regular files and invoke the file command on the results, run.


```
19706@ip-172-26-2-101:~$ find . -type f -exec file '{}' \;
./.vim/.netrwhist: ASCII text
./.cache/motd.legal-displayed: empty
./nam.txt: ASCII text
./name.txt/file.txt: ASCII text
./file.txt: empty
./.viminfo: ASCII text
./.bash_history: UTF-8 Unicode text
19706@ip-172-26-2-101:~$
```

ii) To find regular files and display their attributes using the ls command, run.


```
19706@ip-172-26-2-101:~$ find . -type f -exec ls -l '{}' \;
-rw-r--r-- 1 19706 cs230 191 Mar 17 16:00 ./.vim/.netrwhist
-rw-r--r-- 1 19706 cs230 0 Mar 17 16:01 ./.cache/motd.legal-displayed
-rw-r--r-- 1 19706 cs230 566 Mar 17 15:59 ./nam.txt
-rw-r--r-- 1 19706 cs230 567 Mar 17 16:01 ./name.txt/file.txt
-rw-r--r-- 1 19706 cs230 0 Mar 17 15:46 ./file.txt
-rw------- 1 19706 cs230 2132 Mar 17 16:01 ./.viminfo
-rw------- 1 19706 cs230 2272 Mar 17 16:00 ./.bash_history
19706@ip-172-26-2-101:~$
```

iii) To find files over 20 bytes in size and list them out, run



```
19706@ip-172-26-2-101:~$ find ~ -type f -size +20c -exec ls -hl {} \;
-rw-r--r-- 1 19706 cs230 191 Mar 17 16:00 /home/19706/.vim/.netrwhist
-rw-r--r-- 1 19706 cs230 566 Mar 17 15:59 /home/19706/nam.txt
-rw-r--r-- 1 19706 cs230 567 Mar 17 16:01 /home/19706/name.txt/file.txt
-rw------- 1 19706 cs230 2.1K Mar 17 16:01 /home/19706/.viminfo
-rw------- 1 19706 cs230 2.3K Mar 17 16:00 /home/19706/.bash_history
19706@ip-172-26-2-101:~$
```