Q1 . The main function here, had two arguments (int argc, char *argv[]), explain their use?

Ans. Main function always has two arguments by default(or requires two arguments in C), first is the number/ count of arguments and second is argument vector which contains the number of arguments the user gives in the terminal.

For instance in the code I can run it like :
```
./cpu<filename> “first argument” “second argument”

```
And therefore the argc will be 2 and argument vector will contain “first argument” “second argument”.

Q2: How do we increase the delay in printing strings?

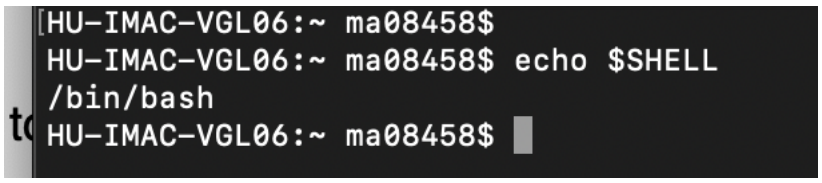
Ans: We can increase the delay by sleep function, or in my case as it was not working on mac so I used a for loop running 1000 * sleep_seconds times to cause a delay after printing a string.

Q3: What is a shell, and which shell are you in?

Ans : A shell is a program that provides us interface to interact with the Operating system like bash.
To check in which shell we are , we run

```
**echo $SHELL**

```



Q4: What is a “Home Directory,” and what is your Home Directory?

Ans: The default directory in which the files are typically saved and is accessed when opening a new terminal , and every command more or less of the terminal works in this directory, to check the home directory we run
```
echo $HOME
```

```
HU-IMAC-VGL06:~ ma08458$ echo $HOME
/Users/ma08458
HU-IMAC-VGL06:~ ma08458$
```

Q5: What's a Working Directory and which directory are you in?

Ans: The current directory we are working in (for instance we are running some commands in a specific folder using BASH) is the working directory
We can check it via

```
pwd
```

```
/Users/ma08458
HU-IMAC-VGL06:~ ma08458$ pwd
/Users/ma08458
HU-IMAC-VGL06:~ ma08458$
```

Q6: Differentiate between an 'Absolute Path' and a 'Relative Path'?

Ans: The absolute path is the path from the root directory while relative path is the path relative or the path starting from and including its parent folder.
So for example a relative path of a file named "somefile.py" in folder "a folder" will be referred as :

```
a folder/somefile.py
```

However for the same file the absolute path will be :

```
/Users/ma08458/Desktop/lab_01/a file/somefile.py
```

Q7: What's the largest file inside the directory "/usr/bin"?

Ans: We can easily find the largest file using the command :

```
ls -1S /usr/bin/ | head -n -2
```

(-1 lists down directories on separate lines)

```
hussainronaque@Hussains-MacBook-Air ~ % ls -1S /usr/bin | head -n 2
parl
parl5.30
hussainronaque@Hussains-MacBook-Air ~ %
```

Parl would be biggest in size

Q8: What's the most recently created file inside the directory /usr/bin?

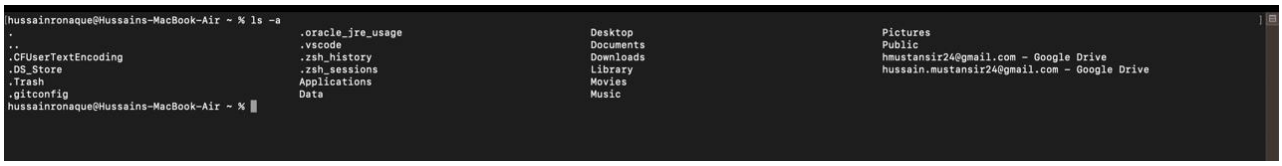
Ans: All files are created on same date when I was checking files in macOS whereas, we can check the most recently added file by running the following command :

```
ls -1t /usr/bin/ | head -n -2
```

Note: here 2 means that it will get the top two lines , of which the 2nd line is the most recent file. Moreover this command displays the file in descending order with respect to time.

Q9: List all the hidden files and directories in your home directory.

Using the ls –a command we can see the hidden directories. The hidden files are the ones that start with `.`



Q10: What does the command 'file' do?

Ans: The command FILE examines the file content to check the file type, (note it does not check the extension but the content inside)

If I run

```
file /bin/ls
```

The output is :

This indicates that file in bin / ls is and executable file type

Q11: Search for the “-h” option of “ls.” What do they do? Use them.

Ans: -h displays the file sizes in a form that is understandable by us humans, to find the command I ran man

```
ls –help
```

And then found :

To run the following command :

```
ls -1h ~
```

Here the command is used with -1 and h where h is command to make file sizes in human understandable form where as -1 is for listing files on separate lines

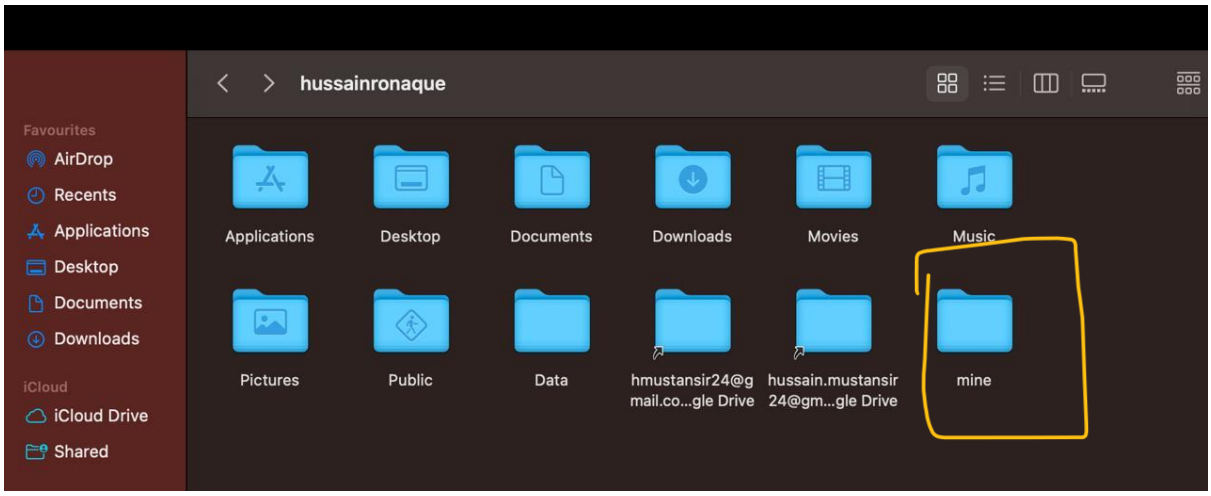
Q12: Make the directory “mine/subdir/subsubdir” using one command only.

Ans: We can make a directory / folder using mkdir command as follows :

```
    mkdir -p ~/mine/subdir/subsubdir
```

Here -p is for parent folder which ensure that the parent folder, if not exists is also created.

Output is :



Q13: While staying in your home directory, create an empty file dummy.txt in mine/subdir/subsubdir.

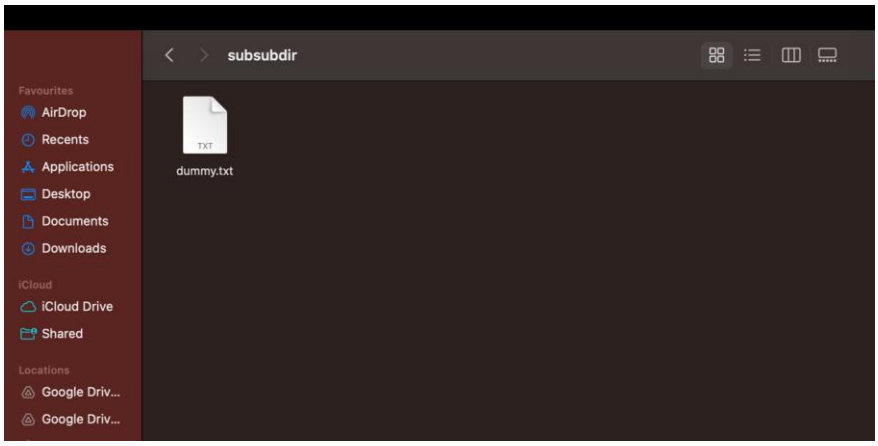
Ans: We can make an empty directory using touch command as follows :

```
touch ~/mine/subdir/subsubdir/dummy.txt
```

This will create an empty file if it does not exists with name dummy.txt :

```
hussainronaque@Hussains-MacBook-Air ~ % touch ~/mine/subdir/subsubdir/dummy.txt
hussainronaque@Hussains-MacBook-Air ~ %
```

(note : mkdir creates directories, while touch creates empty files or updates the timestamps of existing files)

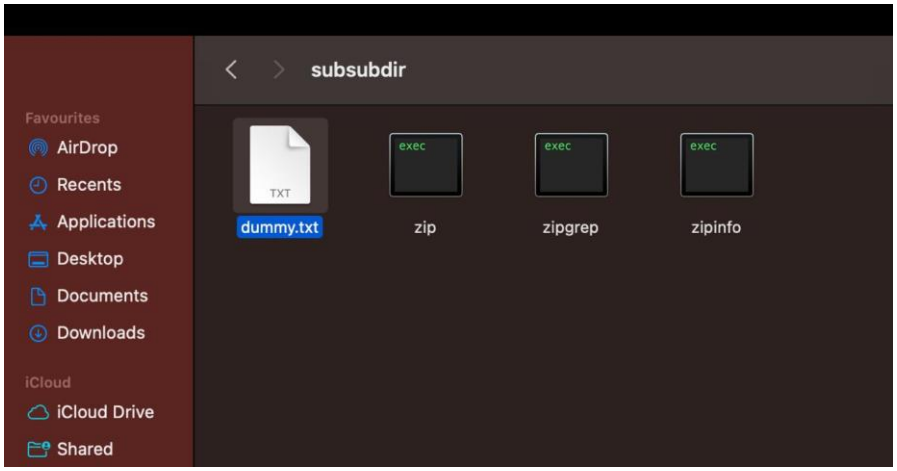


Q14: While staying in your home directory, copy the files zip, zipgrep, zipinfo from /usr/bin to mine/subdir/subsubdir

Ans : To copy the files we can use the command cp as follows :

```
cp /usr/bin/zip /usr/bin/zipgrep /usr/bin/zipinfo ~/mine/subdir/subsubdir/
```

```
hussainronaque@Hussains-MacBook-Air ~ % touch ~/mine/subdir/subsubdir/dummy.txt
hussainronaque@Hussains-MacBook-Air ~ % cp /usr/bin/zip /usr/bin/zipgrep /usr/bin/zipinfo ~/mine/subdir/subsubdir/
hussainronaque@Hussains-MacBook-Air ~ %
```



Q15: Move all files from mine/subdir/subsubdir to mine/subdir/

Ans : We can move the files from one folder to other using mv command as follows :

```
mv ~/mine/subdir/subsubdir/* ~/mine/subdir/
```

(here * in the end of source path indicates to copy the files of all types)

Q16: List all the files in /etc whose second letter is c.

Ans: We can use ls command with a filter to list down all the files with second letter c

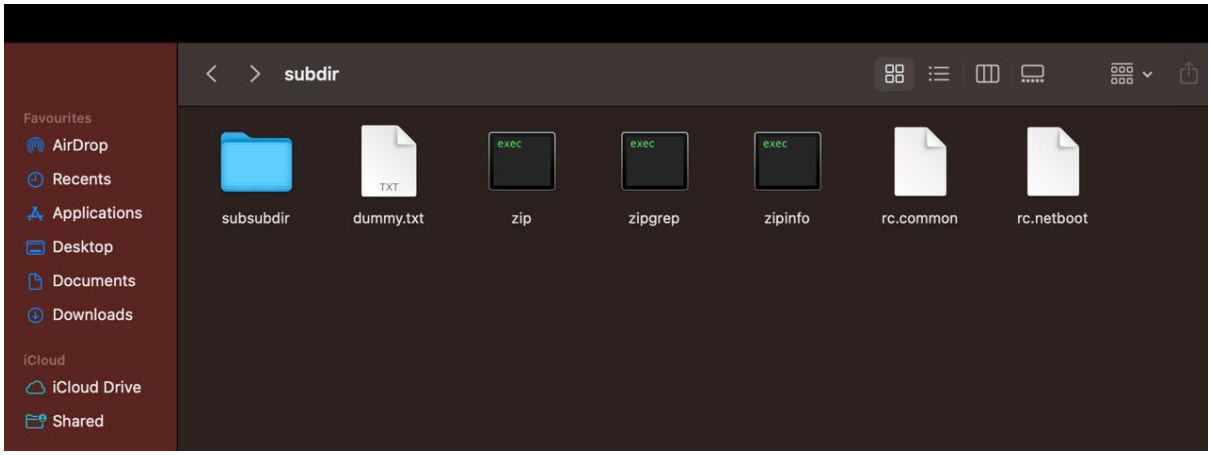
```
    ls /etc/?c*
```

So question mark means any one character before c and anything after c making c at the 2nd position.

Q17: Copy all of them to mine/subdir. Then delete all files that contain a digit.

Ans: We can do this via series of commands :

```
        cp /etc/?c* ~/mine/subdir/
```



Then to remove the files we run :

```
        rm ~/mine/subdir/*[0-9]*
```

Q18: Delete the mine/subdir/ directory

We can again use rm command to remove the files /directory

```
         rm -r /mine/subdir/
```

The -r ensures all directories alongwith their contents are deleted.

```
hussainronaque@Hussains-MacBook-Air ~ % rm ~/mine/subdir/*[0-9]*
zsh: no matches found: /Users/hussainronaque/mine/subdir/*[0-9]*
hussainronaque@Hussains-MacBook-Air ~ % rm -r /mine/subdir/
rm: /mine/subdir/: No such file or directory
hussainronaque@Hussains-MacBook-Air ~ % rm /mine/subdir/
rm: /mine/subdir/: No such file or directory
hussainronaque@Hussains-MacBook-Air ~ % rm /mine/subdir
rm: /mine/subdir: No such file or directory
hussainronaque@Hussains-MacBook-Air ~ % rm mine/subdir/
rm: mine/subdir/: is a directory
hussainronaque@Hussains-MacBook-Air ~ % rm -r mine/subdir/
hussainronaque@Hussains-MacBook-Air ~ %
```