Write a fully-documented class named `BashTerminal`. The class should contain a single `main` method which allows a user to interact with a file system implemented by an instance of `DirectoryTree` using the following commands (note that commands are case-sensitive and will always be lower-case):

Command	Description
`pwd`	Print the "present working directory" of the cursor node (e.g `root/home/user/Documents`).
`ls`	List the names of all the child directories or files of the cursor.
`ls -R`	Recursive traversal of the directory tree. Prints the entire tree starting from the cursor in *pre-order* traversal.
`cd {dir}`	Moves the cursor to the child directory with the indicated name (Only consider the *direct* children of the cursor).
`cd /`	Moves the cursor to the root of the tree.
`mkdir {name}`	Creates a new directory with the indicated name as a child of the cursor, as long as there is room.
`touch {name}`	Creates a new file with the indicated name as a child of the cursor, as long as there is room.
`exit`	Terminates the program.

It should be noted that these commands should all have the same effect as if you were to execute them in a live bash shell on any Unix-based operating system (linux, mac, etc.). However, the command `ls -R` has been modified to make the assignment easier, and you are not expected to move up directories or change directories for absolute and relative paths with the `cd` command. You are encouraged to try these commands on a live bash terminal if you have access to get a feel for how they should work.

- **`public static void main(String[] args)`**
  - ***Brief:***
    - **Runs a program which takes user input and builds a `DirectoryTree` using the commands indicated above.**

EXTRA CREDIT - OPTIONAL

For extra credit, you may include the following features in your submission. These parts are NOT REQUIRED to recieve full credit on your assignment; however, they may be included in your submission if you wish to attempt them.

- **Include the following additional commands in your `BashTerminal`:**

Command	Description
`find {name}`	Finds the node in the tree with the indicated name and prints the path.
`cd ..`	Moves the cursor up to its parent directory (does nothing at root). (e.g. `cd root/home/user/Documents`)
`cd {path}`	Moves the cursor to the directory with the indicated **path**. (e.g. `cd root/home/user/Documents`)
`mv {src} {dst}`	Moves a file or directory specified by src to dst, including all children. (Note that `src` and `dst` are **absolute** paths).

- **Instead of using a ternary (3-child) tree structure, implement `DirectoryTree` so that any node may contain up to 10 child references. If you attempt this, try to avoid having 10 member variables for each `DirectoryNode` (i.e. `child1, child2, child3` etc.). What would be a better way to handle an arbitrary number of child references?**

Sample Input/Output:

NOTE: When prompting user input, you should indicate the name of the user and the host being accessed in the following format:

```
[user@host]: $ // Waiting for command.
```

To assist with grading, please replace 'user' with your own netID. You may feel free to name your host whatever you would like, as long as it is appropriate. Computer Science puns and obscure references will be appreciated.

// Comment in green, input in red, output in black

```
// General use.
Starting bash terminal.
[user@host]: $ pwd
root
[user@host]: $ mkdir dev
[user@host]: $ mkdir home
[user@host]: $ mkdir bin
[user@host]: $ ls
dev home bin
[user@host]: $ cd dev
[user@host]: $ pwd
root/dev
```

```
[user@host]: $ touch ttys0
[user@host]: $ touch ttys1
[user@host]: $ ls
ttsy0 ttsy1
[user@host]: $ cd /
[user@host]: $ pwd
root
[user@host]: $ cd bin
[user@host]: $ touch sublime
[user@host]: $ touch gcc
[user@host]: $ cd /
[user@host]: $ ls -R

|- root             // Note directories begin with '|-'
    |- dev
        - ttys0     // Note files begin with '-'
        - ttys1
    |- home
    |- bin
        - sublime
        - gcc

[user@host]: $ cd home
[user@host]: $ mkdir user
[user@host]: $ cd user
[user@host]: $ pwd
root/home/user
[user@host]: $ mkdir Documents
[user@host]: $ mkdir Pictures
[user@host]: $ mkdir Downloads
[user@host]: $ cd Documents
[user@host]: $ touch hw5.java
[user@host]: $ touch resume.pdf
[user@host]: $ ls
hw5.java resume.pdf
[user@host]: $ cd /
[user@host]: $ cd home
[user@host]: $ cd user
[user@host]: $ cd Pictures
[user@host]: $ touch puppies.jpg
[user@host]: $ cd /
[user@host]: $ ls -R

|- root
    |- dev
        - ttys0
        - ttys1
    |- home
        |- user
            |- Documents
                - hw5.java
                - resume.pdf
            |- Pictures
                - puppies.jpg
            |- Downloads
    |- bin
        - sublime
        - gcc

[user@host]: $ cd home
[user@host]: $ cd user
[user@host]: $ pwd
root/home/user
[user@host]: $ ls -R

|- user
    |- Documents
         - hw5.java
         - resume.pdf
    |- Pictures
         - puppies.jpg
    |- Downloads

[user@host]: $ exit
Program terminating normally


// Special cases.
...
[user@host]: $ pwd
root/dev
```

```
[user@host]: $ ls -R

|- dev
     - ttys0
     - ttys1
     - ttys2

[user@host]: $ touch ttys3
ERROR: Present directory is full.
[user@host]: $ cd ttys2
ERROR: Cannot change directory into a file.
[user@host]: $ cd nonexistantDirectory
ERROR: No such directory named 'nonexistantDirectory'.
[user@host]: $ exit
Program terminating normally


// EXTRA CREDIT - NOT REQUIRED.
...
[user@host]: $ pwd
root
[user@host]: $ ls -R

|- root
    |- dev
         - ttys0
         - ttys1
    |- home
        |- user
            |- Documents
                |- hw5.java
                |- myFolder
                     - file1.txt
                     - file2.txt
                     - file3.txt
            |- Pictures
                 - puppies.jpg
    |- tmp
         - puppies.jpg

[user@host]: $ find puppies.jpg
root/home/user/Pictures/puppies.jpg
root/tmp/puppies.jpg
[user@host]: $ find kittens.jpg
ERROR: No such file exits.
[user@host]: $ cd home/user // Note cd with path.
[user@host]: $ pwd
root/home/user
[user@host]: $ cd .. // Move up to parent.
[user@host]: $ pwd
root/home
[user@host]: $ cd ..
[user@host]: $ pwd
root
[user@host]: $ cd ..
ERROR: Already at root directory.
[user@host]: $ pwd
root
[user@host]: $ mv root/home/user/Documents/myFolder root/tmp // Note absolute paths.
[user@host]: $ ls -R

|- root
    |- dev
         - ttys0
         - ttys1
    |- home
        |- user
            |- Documents
                 - hw5.java
            |- Pictures
                 - puppies.jpg
    |- tmp
        |- myFolder
             - file1.txt
             - file2.txt
             - file3.txt

[user@host]: $ exit
Program terminating normally
```
