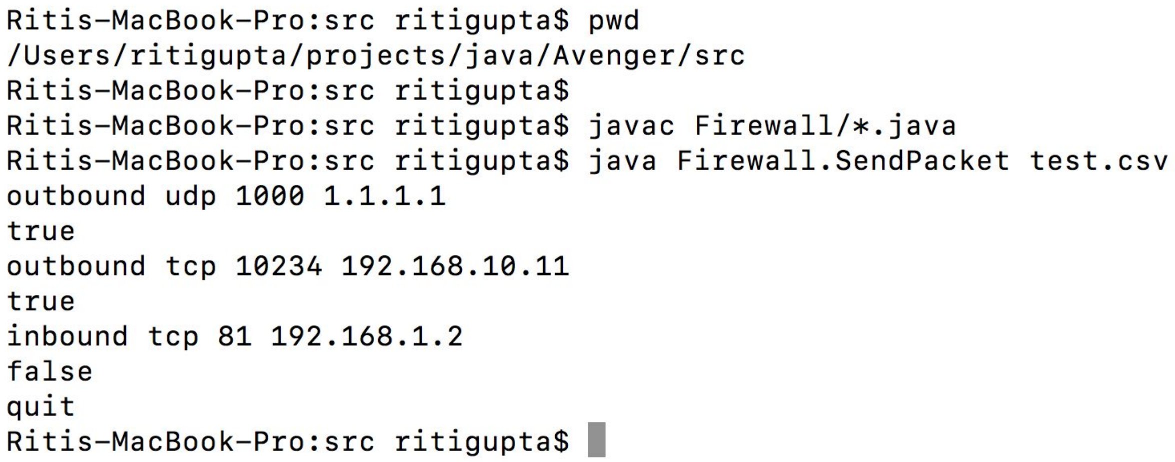
How to run

- Go inside src folder
- javac Firewall/*.java
- java Firewall.SendPacket “path to csv file”
  Enter space separated 4 values and then press “ENTER”
  dir protocol port ipaddress
- “quit” will end

SAMPLE OUTPUT



```
Ritis-MacBook-Pro:src ritigupta$ pwd
/Users/ritigupta/projects/java/Avenger/src
Ritis-MacBook-Pro:src ritigupta$
Ritis-MacBook-Pro:src ritigupta$ javac Firewall/*.java
Ritis-MacBook-Pro:src ritigupta$ java Firewall.SendPacket test.csv
outbound udp 1000 1.1.1.1
true
outbound tcp 10234 192.168.10.11
true
inbound tcp 81 192.168.1.2
false
quit
Ritis-MacBook-Pro:src ritigupta$ 
```

CODE FLOW :

- CSV file passed to constructor is parsed and stored in the database in memory
- Reading the file again and again from the disk every time is time consuming and hence stored in the database
- Main method is in SendPacket class
- The file is stored in a matrix where each row is for direction and each column is for protocol. The matrix stores list of <port,address> specified in rule. This reduces the space as there would be one list for each direction and protocol. For example, in this assignment, there would be 4 lists stored as follows
  Mat[0][0] = InboundUDPlist
  Mat[1][0] = OutboundUDPlist
  Mat[0][1] = InboundTCPlist
  Mat[1][1] = OutboundUDPlist

The matrix in the code is indexed based on the enum values. Have added flexibility to add protocols and directions in future. The size of matrix would get increased automatically when enum fields would be added

- I have added enum values to make the code easy to understand and make it possible to add more protocols and directions in future.

TEST METHODOLOGY:

- First validated with the values given in the coding assignment
- Tested for large values of port range and address ranges
- Tested for starting of the range specified
- Tested for outside the range
- Increased the number of rules to 100000 manually by repeating many of them, runs fine
- More testing if had more time
- Included "test.csv"

IMPROVEMENTS REQUIRED

- Duplicate rules should be detected and not stored multiple times. This would memory and time both as less number of rules would be compared to validate the packets.
- If overlaping ranges are specified , should be able to merge them to save space and time both.
- Since the octet values are in 0-255, could have unsigned char instead of a string which would have saved space.
- Validate the data and get rid of it before matching with any rule, if corrupt(negative) input specified, this would again save time.

OTHER

- I have saved the ranges in a pair as the ranges might be large 0 - 255. Storing for each value (0,1,2,....,255) would have improved time efficiency but memory would increase drastically.

TEAMS INTERESTED :

I am open to working for any of the teams specified. All the roles specified align with my interest and skillset. I am planning to take Distributed Computing and Cloud computing in next semester which should be helpful as well as per the descriptions provided. I would be able to understand better after knowing more about the teams.