DATA STRUCTURE AND ALGORITHMS.

ASSIGNMENT
REPORT
ON

“SOCIAL SIMULATION”

AUTHOR

RAJ SURESH PATEL

STUDENT ID

19701855

PROF.DR. VALERIE MAXVILLE

TABLE OF CONTENT

1. Abstract
   1.1 Some recommendations for the program

2. Background

3. Methodology
   3.1 Choice of Parameter

4. Results

5. Conclusion

1. Abstract:-

The primary purpose of this report is to discuss the idea and logic behind the project so-called SocialSimulation, which was created to answer the assignment conundrum. Included goal contents creating a lookalike social media network such as Facebook or Instagram. This simulation module will allow user or admin to handle the system in a manner in which admin will be able to set and change some parameters, including the probability of liking a post and following a profile. As per assignment specification has all the functionalities of a working social media network such as adding/removing people profile, people posting and supporting other people and finally people getting like from their followers on their post. This module will showcase and give three functionalities in which the code can change its behaviour of executing, for example, (D)efault (I)nteractive and lastly, (S)imulation. No in-built functionalities were used during the implementation of this code, though some exception which was allowed such as .split() and .readlines(). All system is based on the practical assignment 3 and 5 of "Data Structure and Algorithms" and will use the classes from this possible. This is purely based on a linked list(doubly ended), list node, and the graphs. Lastly, two main functionalities were added, such as stats and running events in a timestep which gives a clear definition or a bright look at how the network is functioning. Also, to make sure that all the functionalities are working inside the classes, a test harness was written to check each function in the step by step manner, which shows that the code is working fine.

1.1 Some recommendations for the program:-

- An added functionality for people to delete their post or update any previous job.
- Giving a visual representation of an overall network using matplotlib.
- Deleting profile that has been not updating, following or liking any post or pattern from a prolonged period or say in at least 3 to 4 timesteps.
- Statistics may vary according to a different value for which to tackle and stabilise the module should run a varied number of parameters for each network before any update into it.
- To save the network on the updated network file will be saved to check and compare how much original vs new record differed after running timesteps based on probabilities.

2. Background:-

The purpose of this program is as follow:-

- To do a code module which can be used for the analytical meaning of visualising a network in the form of the linked list and a graph, this network represents a social simulation like facebook or Instagram by which one can adhere all the analysis of a real-world working social media and how its function based on given probabilities.

- This program has the capability of output a network which is saved after running the timesteps based on the probability for better analysis of it.

- This data can help exceptionally in the field of newly emerging social sites to list out all the famous person who has most numbers of likes and followers and check the statistics along with the time to how a particular profile gets up in the top for most liked and followed account.

- This module of social simulation can also enable its user to check the likelihood of people getting followed by someone after he/she had like a post from that person indirectly.

- The program enables users to enter the data for the new post, make people follow other people from command line argument to check how it changes the overall network.

- This approach for creating the code for simulation shows the network in one timestep and will also list out all the links and post made by people each.

- Each timestep will create some sort of statistics which will be outputted as a graph to check the operationality of the overall network.

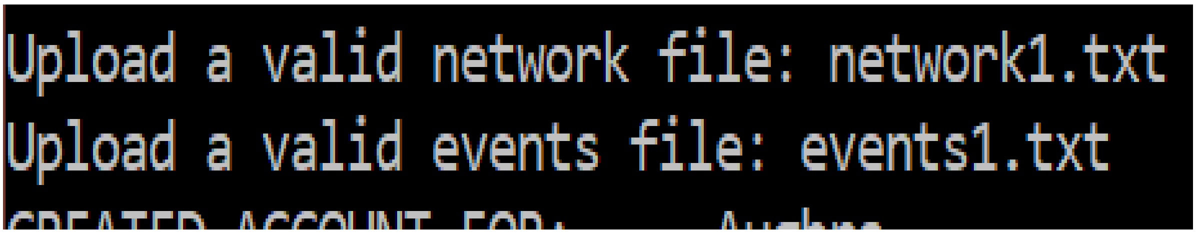
3. Methodology:-

3.1 Choice of Parameter:-

- Parameters such as loading the network file which will be converted into accounts and also link per se following/followed by was firstly were split into two-part to read and then to process it accordingly.
- The above symbol represents there true meaning in the provided network and events file.
  : = FOLLOWING
  F = FOLLOWS(F:A: B – >A FOLLOWS B)
  A = ADD NEW ACCOUNT (A: NEW – WILL ADD NEW ACCOUNT CALLED NEW)
  U = UNFOLLOW (U:A:B -> A UNFOLLOWED B)
  P = POST(P:A:HII -> A POSTED HII)
  R = REMOVE(R:A -> PROFILE A WAS REMOVED)
- Assigning the initial probabilities of liking and following a post which should result in between 0.1 to 1.0 as random.random() function is used to examine the behaviour in the network and how does it affect the following ratios.
- Performance-based on the result was expected to be showing all the network inside flow such as a linked list

3.2 Used Technique:-

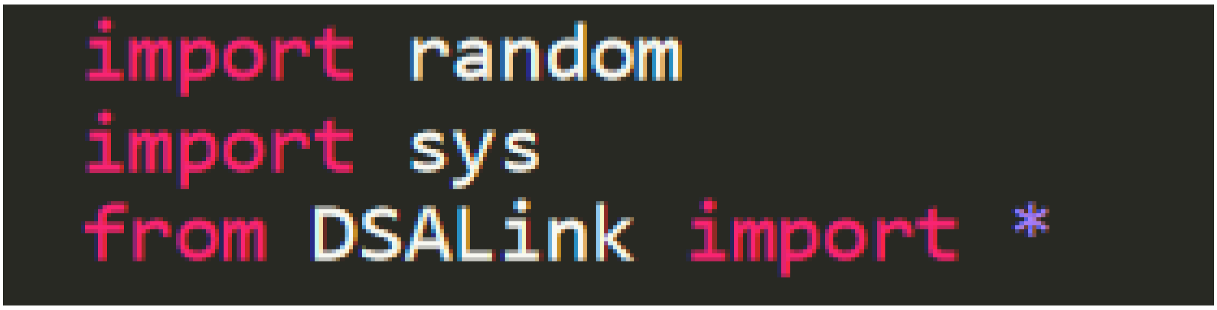
- In this project, the user can pass the parameter such as files name of network file and event file to load into the network. This module of social simulation will list out all the connections such as people in the system, what they posted, whom they are following and unfollowing.
- Some functionalities are as shown in the above figure.



```
Upload a valid network file: network1.txt
Upload a valid events file: events1.txt
CREATED ACCOUNT FOR:      [illegible]
```

Figure 3.2.1 – Uploading the network and events file via command line argument

- Some library and files were imported which were allowed to use in the code to just check some functionalities, they are as shown below:

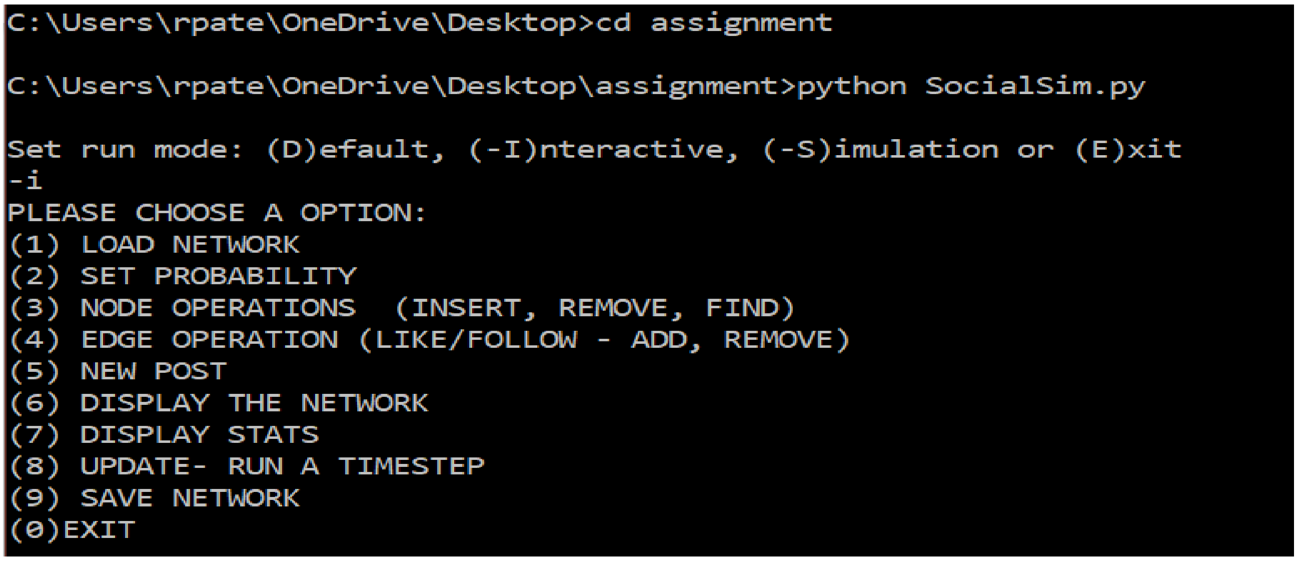


```python
import random
import sys
from DSALink import *
```

Figure 3.2.2 – Importing needed libraries and based files. (From previous practical)

- A method such as create a post, get likes, get follower, follower and following were implemented to make a function for getting and posting a job from the people profile ends. Where this post can acquire likes and may get a follow on their profiles which will be helpful.
- Also, a function or per se a class was created separately for Node operations and edge operations where node operations have the functioning of Inserting, removing and finding a profile and its connection.
- Three main classes, for example, NodeFunc and EdgeFunc and network, is created to hold separate classes and were called inside other courses by using NodeFunc(), EdgeFunc() and system ().

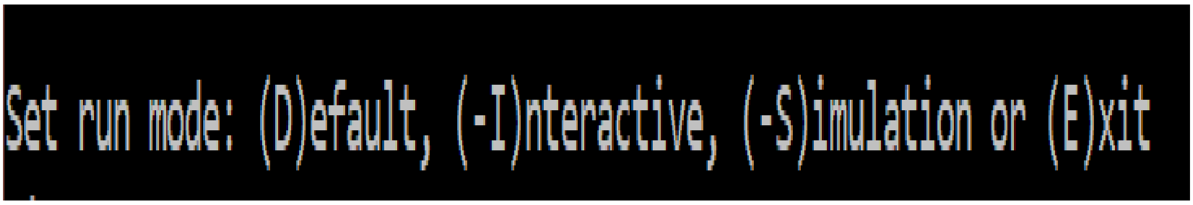
- A drop-down list like scenario was written to enable to make a decision based on the list. This menu will give the user many functionalities ranging from loading any network file to the network. All the operation can be performed form this menu, which is shown below:-



```
C:\Users\rpate\OneDrive\Desktop>cd assignment

C:\Users\rpate\OneDrive\Desktop\assignment>python SocialSim.py

Set run mode: (D)efault, (-I)nteractive, (-S)imulation or (E)xit
-i
PLEASE CHOOSE A OPTION:
(1) LOAD NETWORK
(2) SET PROBABILITY
(3) NODE OPERATIONS  (INSERT, REMOVE, FIND)
(4) EDGE OPERATION (LIKE/FOLLOW - ADD, REMOVE)
(5) NEW POST
(6) DISPLAY THE NETWORK
(7) DISPLAY STATS
(8) UPDATE- RUN A TIMESTEP
(9) SAVE NETWORK
(0)EXIT
```

Figure 3.2.3:- Menu List for the operation of the network

- Three basic functionalities like default, interactive and simulation mode were defined to run the overall module in their respective environment.
- Class post has the function like creating a DSALinkedList for saving and updating post function, it also has total likes and gets post function which give extra operationality like getting the total number of likes a profile has and how many post-its did from the starting of the network.
- The command-line argument can be pass from the CMD to execute the code blocks as per user-defined inputs exteriorly.



```
Set run mode: (D)efault, (-I)nteractive, (-S)imulation or (E)xit
```

Figure 3.2.4 – CMD operations for default, interactive and simulation mode.

3.3 Complexity Analysis:-

Two different display function can be used, which is displaying the graph as a list and display as a matrix, where its depend on the criteria that are based on valid parameters. Concerning time and space complexity displaying a graph as a list has least complexity as compared to that of presenting it as a matrix.

As shown in the above differentiation between the displaying the adjacency list as a list and a matrix will clear out the doubt for why this parameter was chosen.

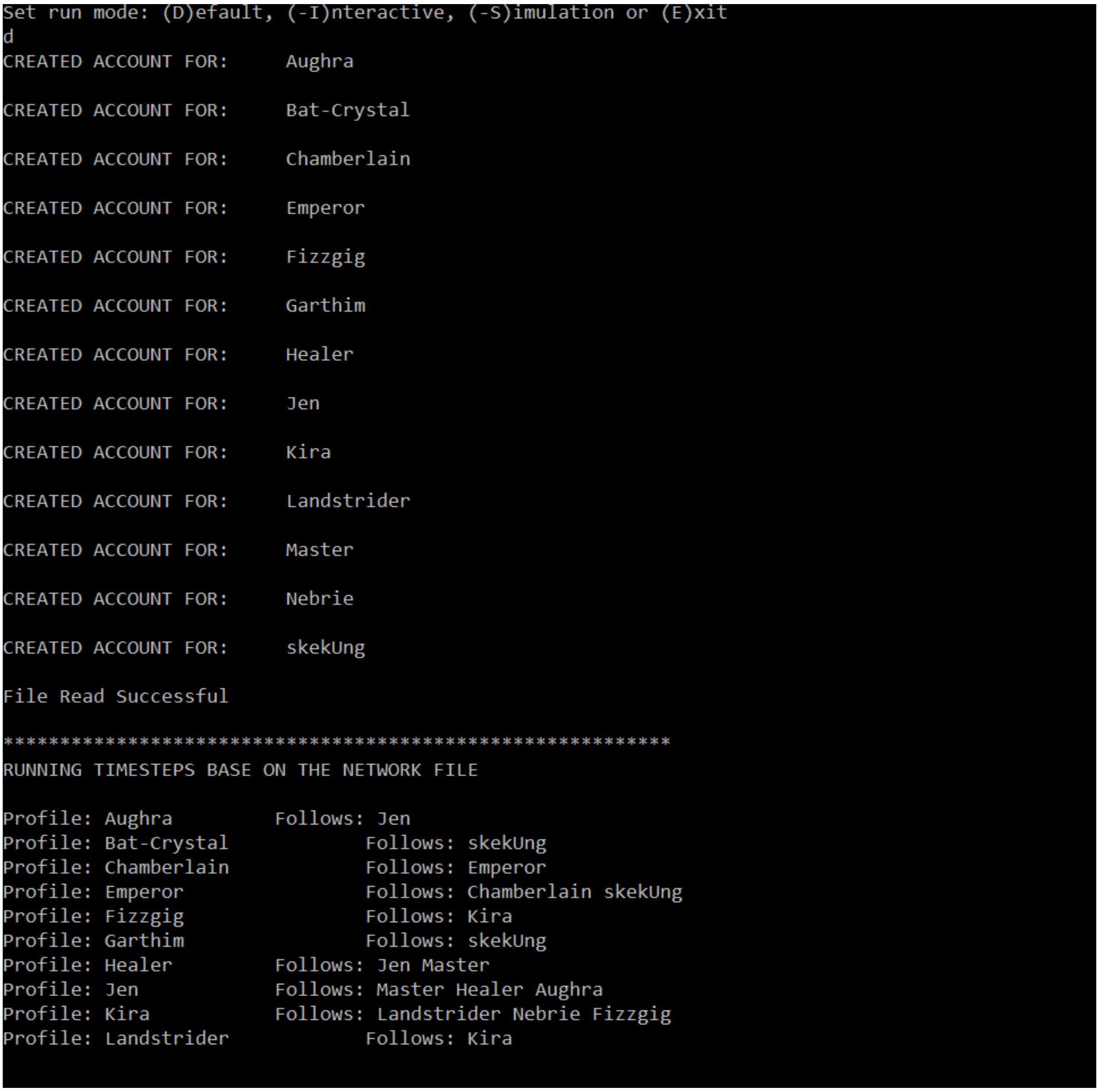
As Adjacency list, while displaying it as a list which iterates significantly faster as compared to that of a matrix does.

The adjacency list is quicker in inserting, removing and searching while adjacency list is comparatively slower as compared to the adjacency matrix.

4. Result:-

The output for the simulation is as follow with each timestamp.

When all the input were inputted from the previously given data files such as network1.txt and network2.txt with its sub-file which are events1.txt, event2.txt.



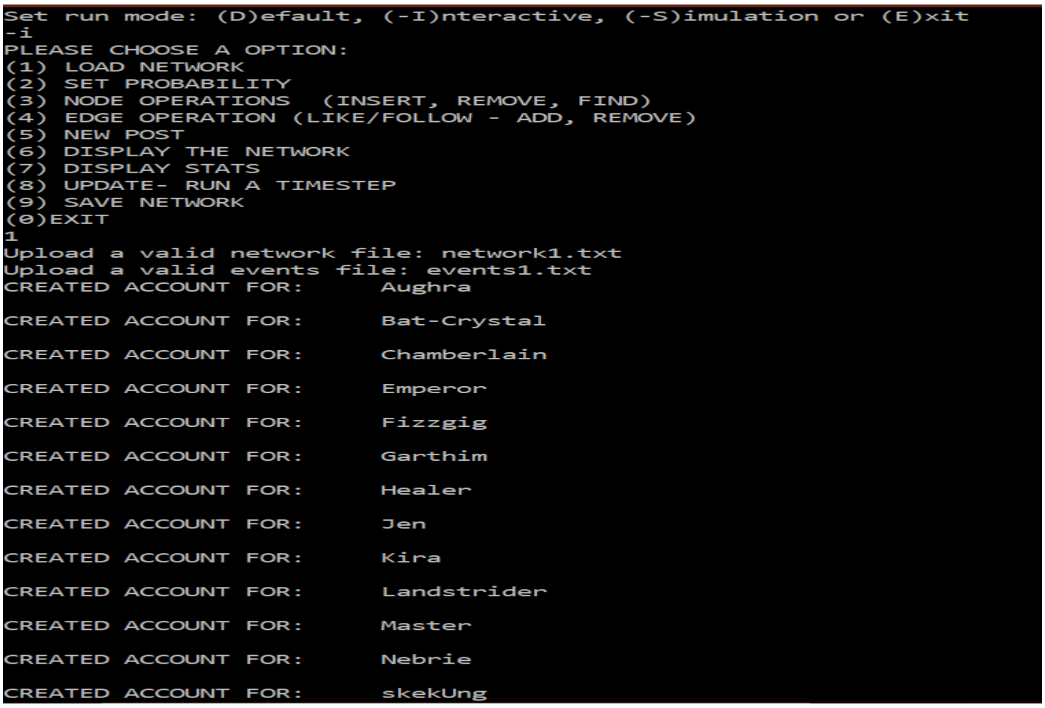
```
Set run mode: (D)efault, (-I)nteractive, (-S)imulation or (E)xit
d
CREATED ACCOUNT FOR:     Aughra

CREATED ACCOUNT FOR:     Bat-Crystal

CREATED ACCOUNT FOR:     Chamberlain

CREATED ACCOUNT FOR:     Emperor

CREATED ACCOUNT FOR:     Fizzgig

CREATED ACCOUNT FOR:     Garthim

CREATED ACCOUNT FOR:     Healer

CREATED ACCOUNT FOR:     Jen

CREATED ACCOUNT FOR:     Kira

CREATED ACCOUNT FOR:     Landstrider

CREATED ACCOUNT FOR:     Master

CREATED ACCOUNT FOR:     Nebrie

CREATED ACCOUNT FOR:     skekUng

File Read Successful

***********************************************************
RUNNING TIMESTEPS BASE ON THE NETWORK FILE

Profile: Aughra         Follows: Jen
Profile: Bat-Crystal            Follows: skekUng
Profile: Chamberlain            Follows: Emperor
Profile: Emperor                Follows: Chamberlain skekUng
Profile: Fizzgig                Follows: Kira
Profile: Garthim                Follows: skekUng
Profile: Healer         Follows: Jen Master
Profile: Jen            Follows: Master Healer Aughra
Profile: Kira           Follows: Landstrider Nebrie Fizzgig
Profile: Landstrider            Follows: Kira
```

Figures 4.1:- Output in Default mode with

Network file name – network1.txt

Event File Name – events1.txt

.

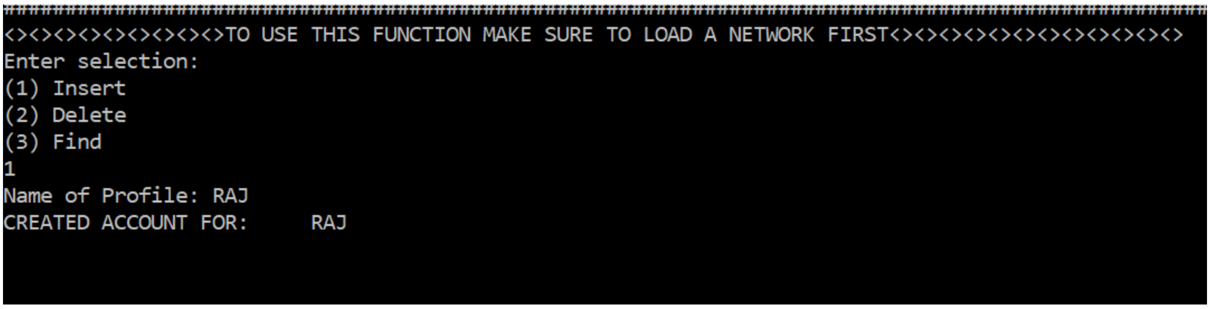


```
Set run mode: (D)efault, (-I)nteractive, (-S)imulation or (E)xit
-i
PLEASE CHOOSE A OPTION:
(1) LOAD NETWORK
(2) SET PROBABILITY
(3) NODE OPERATIONS  (INSERT, REMOVE, FIND)
(4) EDGE OPERATION (LIKE/FOLLOW - ADD, REMOVE)
(5) NEW POST
(6) DISPLAY THE NETWORK
(7) DISPLAY STATS
(8) UPDATE- RUN A TIMESTEP
(9) SAVE NETWORK
(0)EXIT
1
Upload a valid network file: network1.txt
Upload a valid events file: events1.txt
CREATED ACCOUNT FOR:     Aughra

CREATED ACCOUNT FOR:     Bat-Crystal

CREATED ACCOUNT FOR:     Chamberlain

CREATED ACCOUNT FOR:     Emperor

CREATED ACCOUNT FOR:     Fizzgig

CREATED ACCOUNT FOR:     Garthim

CREATED ACCOUNT FOR:     Healer

CREATED ACCOUNT FOR:     Jen

CREATED ACCOUNT FOR:     Kira

CREATED ACCOUNT FOR:     Landstrider

CREATED ACCOUNT FOR:     Master

CREATED ACCOUNT FOR:     Nebrie

CREATED ACCOUNT FOR:     skekUng
```

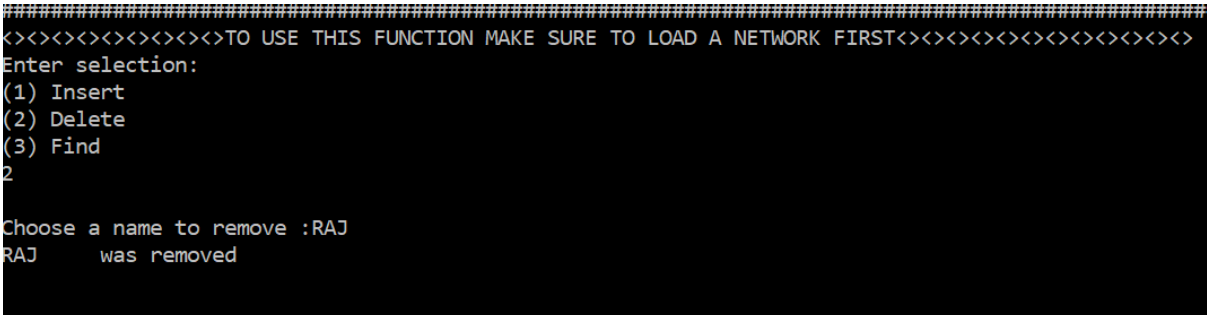
Figures 4.2 :- Interactive Mode(-i)

Network file name – network1.txt

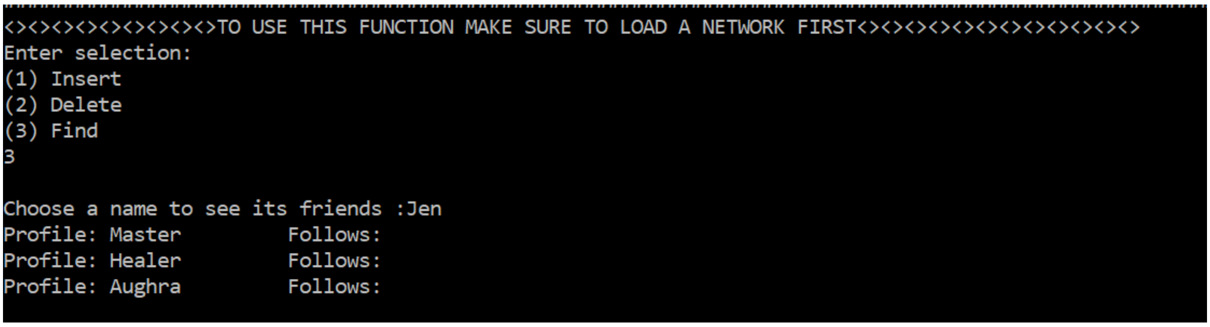
Event File Name – events1.txt



```
################################################################################################
<><><><><><><><><>TO USE THIS FUNCTION MAKE SURE TO LOAD A NETWORK FIRST<><><><><><><><><><><><>
Enter selection:
(1) Insert
(2) Delete
(3) Find
1
Name of Profile: RAJ
CREATED ACCOUNT FOR:     RAJ
```

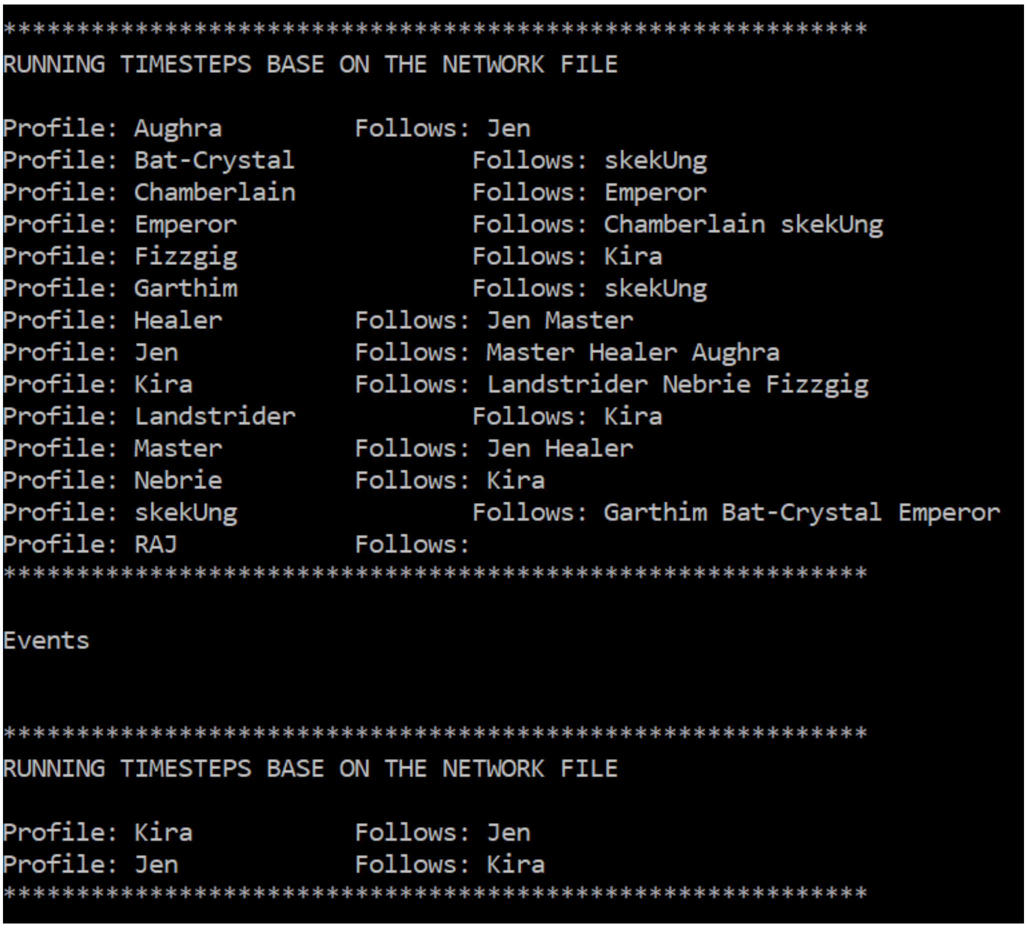


```
################################################################################################
<><><><><><><><><>TO USE THIS FUNCTION MAKE SURE TO LOAD A NETWORK FIRST<><><><><><><><><><><><>
Enter selection:
(1) Insert
(2) Delete
(3) Find
2

Choose a name to remove :RAJ
RAJ     was removed
```



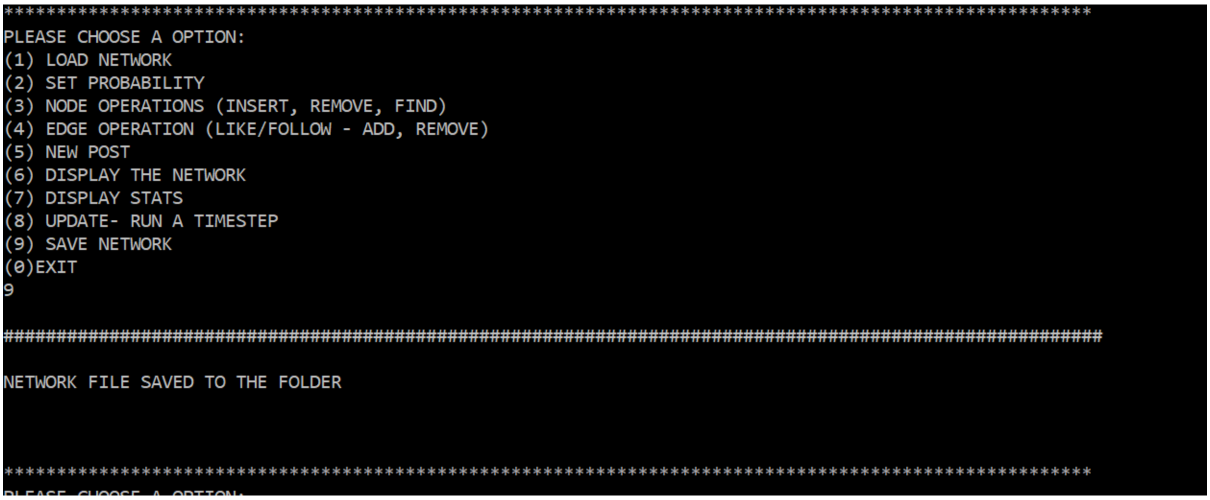
```
<><><><><><><><><><>TO USE THIS FUNCTION MAKE SURE TO LOAD A NETWORK FIRST<><><><><><><><><><><><><><>
Enter selection:
(1) Insert
(2) Delete
(3) Find
3

Choose a name to see its friends :Jen
Profile: Master         Follows:
Profile: Healer         Follows:
Profile: Aughra         Follows:
```

Figures 4.3:- Output for INSERT, DELETE AND FIND function under



```
************************************************************
RUNNING TIMESTEPS BASE ON THE NETWORK FILE

Profile: Aughra         Follows: Jen
Profile: Bat-Crystal            Follows: skekUng
Profile: Chamberlain            Follows: Emperor
Profile: Emperor                Follows: Chamberlain skekUng
Profile: Fizzgig                Follows: Kira
Profile: Garthim                Follows: skekUng
Profile: Healer         Follows: Jen Master
Profile: Jen            Follows: Master Healer Aughra
Profile: Kira           Follows: Landstrider Nebrie Fizzgig
Profile: Landstrider            Follows: Kira
Profile: Master         Follows: Jen Healer
Profile: Nebrie         Follows: Kira
Profile: skekUng                Follows: Garthim Bat-Crystal Emperor
Profile: RAJ            Follows:
************************************************************

Events


************************************************************
RUNNING TIMESTEPS BASE ON THE NETWORK FILE

Profile: Kira           Follows: Jen
Profile: Jen            Follows: Kira
************************************************************
```

Figures 4.4:- Output for Displaying Network



```
********************************************************************************************************
PLEASE CHOOSE A OPTION:
(1) LOAD NETWORK
(2) SET PROBABILITY
(3) NODE OPERATIONS (INSERT, REMOVE, FIND)
(4) EDGE OPERATION (LIKE/FOLLOW - ADD, REMOVE)
(5) NEW POST
(6) DISPLAY THE NETWORK
(7) DISPLAY STATS
(8) UPDATE- RUN A TIMESTEP
(9) SAVE NETWORK
(0)EXIT
9

#########################################################################################################

NETWORK FILE SAVED TO THE FOLDER



********************************************************************************************************
PLEASE CHOOSE A OPTION:
```

Figures 4.5:- Saving file to the folder for the updated network.

5. Conclusion:-

To sum up, this program will allow a network lets say facebook look like which can be used in many sectors to check and analyse a graph and list and how it is functioning based on the given parameter. How probability is changed the stats and also has the functionality of adding, deleting,searching a node(account) form the command line.

Further future work:-

Finally, hash table and trees can be utilised for better visualisation of the network and to run the timesteps. Code can be modified to remove and unfollow someone based on a dislike which will have similar or less than the probability of liking any post.