

News Feed

Your aim is to write a console program that can simulate a social network.

Users have the following capabilities (command to be used in [-]):

- [signup] A user can signup to the system
- [login] A user will be able to login to the system and the session will be created and used across for the other commands. Any existing session will be reinitiated with the new user logins.
- [post] A user can post a feed item.
- [follow] Users can follow other users.
- [reply] A user can comment on another user's feed item.
- [upvote/downvote] Upvote or downvote posts.
- [shownewsfeed] Any user can read his news feed. News items are sorted based on the following:
 - Followed users: posts by followed users appear first.
 - Score (= upvotes - downvotes): higher the better.
 - The number of comments: higher the better.
 - Timestamp: more recent the better.

Bonus:

- Allow users to comment on a comment and upvote/downvote a comment.
- Display time in language like 2m ago, 1 hr ago etc.

Input and output

The input should be taken in the form of statements or driver methods or through a file. The statements should consist of commands and arguments. Commands can include 'signup', 'newsfeed', 'upvote' etc. Arguments can vary depending on the commands.

Example - reply(feed_id, reply_text)

Sample test cases:

Input is in bold, output is not in bold. You need to print the output of *shownewsfeed* for a user, whenever he logs in.

Note: Sample test cases are only for briefing the requirement, the inputs can be taken in the form of statements or driver methods or through a file as per the choice.

```
signup~lucious
signup~albus
signup~tom
login~tom
post~I am going to be the darkest dark wizard of all time
post~I am lord Voldemort btw 3:)
login~lucious
shownewsfeed
    id: 002
    (0 upvotes, 0 downvotes)
    tom
    I am lord Voldemort btw 3:)
    1994-04-19 10:11 PM

    id: 001
    (0 upvotes, 0 downvotes)
    tom
    I am going to be the darkest dark wizard of all time
    1994-04-19 10:10 PM

upvote~001
follow~tom
reply~001~I am with you dark lord!
login~albus
shownewsfeed
    id: 001
    (1 upvote, 0 downvotes)
    tom
    I am going to be the darkest dark wizard of all time
    1994-04-19 10:10 PM
```

```
id: 003
Lucious
I am with you dark lord!
1994-04-19 10:15 PM
```

```
id: 002
(0 upvotes, 0 downvotes)
tom
I am lord Voldemort btw 3:)
1994-04-19 10:11 PM
```

post~Happiness can be found, even in the darkest of times, if one only remembers to turn on the light
follow~tom
downvote~001
downvote~002
reply~002~LOL!
shownewsfeed

```
id: 001
(1 upvote, 1 downvotes)
tom
I am going to be the darkest dark wizard of all time
1994-04-19 10:10 PM
```

```
id: 003
lucius
I am with you dark lord!
1994-04-19 10:15 PM
```

```
id: 002
(0 upvotes, 1 downvotes)
tom
I am lord Voldemort btw 3:)
1994-04-19 10:11 PM
```

```
id: 005
albus
LOL!
1994-04-19 10:30 PM
```

```
id: 004
(0 upvotes, 0 downvotes)
albus
Happiness can be found, even in the darkest of times, if one only remembers to turn on the light
1994-04-19 10:28 PM
```

Guidelines:

- Input can be read from a file or STDIN or coded in a driver method.
- Output can be written to a file or STDOUT.
- Feel free to store all interim/output data in memory.
- Restrict internet usage to looking up syntax
- You are free to use the language of your choice.
- Save your code/project by your name and email it. Your program will be executed on another machine. So, explicitly specify dependencies, if any, in your email.

Expectations:

- The code should be demo-able (very important). The code should be functionally correct and complete.
 - At the end of this interview round, an interviewer will provide multiple inputs to your program for which it is expected to work
- Code should handle edge cases properly and fail gracefully. Add suitable exception handling, wherever applicable.
- Handle concurrency where needed.
- Code should have a good object-oriented design.
- Code should be readable, modular, testable and extensible. Use intuitive names for your variables, methods and classes.
 - It should be easy to add/remove functionality without rewriting a lot of code.
 - Do not write monolithic code.

- Don't use any databases.