

```

struct Tweet {
    int userID;
    int tweetID;

    Tweet(int _u, int _t) {
        this->userID = _u;
        this->tweetID = _t;
    }
};

class Twitter {
    stack<Tweet> newsFeed;
    unordered_map<int, list<int>> following;

public:
    Twitter() {

    }

    void postTweet(int userId, int tweetId) {
        Tweet newTweet(userId, tweetId);
        newsFeed.push(newTweet);
    }

    vector<int> getNewsFeed(int userId) {
        stack<Tweet> tempNewsFeed = newsFeed;
        vector<int> news;
        int ci = 0;
        while (!tempNewsFeed.empty() && ci < 10) {
            Tweet tweet = tempNewsFeed.top();
            tempNewsFeed.pop();

            // is user follows tweeter?
            bool follow = false;

            for (auto x : following[userId]) {
                if (x == tweet.userID) {
                    follow = true;
                    break;
                }
            }

            if (userId == tweet.userID || follow) {
                news.push_back(tweet.tweetID);
                ci++;
            }

        }
        return news;
    }

    void follow(int followerId, int followeeId) {
        following[followerId].push_back(followeeId);
    }
}

```

```
void unfollow(int followerId, int followeeId) {
    following[followerId].remove(followeeId);
}
};

/**
 * Your Twitter object will be instantiated and called as such:
 * Twitter* obj = new Twitter();
 * obj->postTweet(userId,tweetId);
 * vector<int> param_2 = obj->getNewsFeed(userId);
 * obj->follow(followerId,followeeId);
 * obj->unfollow(followerId,followeeId);
 */
```