

# CENG140

## Section I

### C Programming

Spring 2017-2018

### Take Home Exam 2

## 1 Regulations

- Due date: 28 May 2018, Monday, 23:55 Not subject to postpone, please do not ask for.
- Submission: Electronically. You will be submitting a file containing your source code through COW (More detailed description in Grading and Specifications part). Resubmission is allowed (till the last moment of the due date), the last will replace the previous.
- Cheating: We have zero tolerance policy for cheating. People involved in cheating will be punished according to the university regulations.
- Team: There is no teaming up. The take home exam has to be done/turned in individually.
- Newsgroup: You must follow the newsgroup ([news.ceng.metu.edu.tr](https://news.ceng.metu.edu.tr)) for discussions and possible updates on a daily basis.
- Evaluation: Your program will be evaluated automatically using black-box technique so make sure to obey the specifications. Any program that cannot collect 30 points will be graded by eye (glass-box evaluation). These eye evaluation are not open to question. They are final and decisive and are not subject to any objection or questioning.
- Disclaimer: The story of THE2 below is completely fictional, while exhibiting some similarities to some existing systems.

## 2 Introduction

In this homework, you will simulate a discussion platform similar to a forum, but a simpler version. You are expected to use given struct definitions in order to manage the objects in the system and use linked lists to be able to perform the required operations dynamically.

The simulation setup essentially involves users who follow some topics. Each user has a message list that stores the messages to be shown to her when she logs in to the system. Whenever a user sends a message, it is associated with a topic, the message is added to the message list of herself and all the other users who follow that topic. The aim of the simulation is determining the messages that will be seen by each user when they login to the system (that is, at the end of the simulation). At each time step of the simulation, there can be one of the following operations:

- **ADD <username>:** This command adds a new user to the system with the given username. The username is unique. Furthermore, you should assign an integer user id (starting from 1) for each new user. Each user has a message list and subscribed topics list that are initially empty.
- **SUBSCRIBE <username> <topicname>:** This command should add the user id of user to the follower list of topic with topicname. If a topic with the given topicname does not exist, a new topic with given topicname will be created. Each topic should have a topic id, starting from 1.
- **SEND <username> <topicname>:** Each time you encounter a SEND message you will assume there is a new message associated with a message id, author and a topic. The message ids are generated internally for all the messages stored in the system (we don't store the message text for simplicity), starting with the id 1 that is assigned the first message. The author field of the message should be set to the user id of the sender and the topic field should be set to the topic id of the topic. The message (with the id and author) should be added to the message list of both the sender and all users who are following the topic. However, if the sender is a follower of the topic, the message should appear in her message list only once.  
  
The message lists must be always kept sorted according to the message ids, from the earliest message (i.e., with the smallest id) to the most recent one. The topic with the given topicname will always be existent.
- **UNSUBSCRIBE <username> <topicname>:** This command should remove the user id of user from the follower list of topic. Furthermore, all the messages received from that topic in the message list of user will be deleted.
- **DELETEUSER <username> :** This command should remove the user from the list of users. Furthermore, all the messages that are sent from the user will be deleted from other users' inbox.

### 3 Grading and Specifications

You should use struct definitions given below.

```
struct node {
    int data;
    struct node *next;
};

struct message_node{
    int sender_id;
    int topic_id;
    int message_id;
    struct message_node *next;
};
```

```

struct topic_node{
    char *topic_name;
    int topic_id;
    struct node *follower_list; //hold user ID's in data field
    struct topic_node *next;
    //Add additional fields as needed
};

struct user_node{
    char *user_name;
    int user_id;
    struct node *followed_topics; //hold topic ID's in data field
    struct message_node *inbox;
    // Add additional fields as needed
};

```

- You are expected to implement you solution using linked lists. Otherwise you will not be able to collect points.
- Usernames and topic names will consist of at most 20 characters.
- Usernames and topic names will not contain any spaces.
- Obey I/O format strictly. Otherwise you will not be able to collect points since the exam will be graded by blackbox technique.
- Name your source code as the2.c.

## 4 Sample Input and Output

Your code will parse each line in the simulation and run the corresponding command. At the end of the simulation, for each user (in the order of user id) you will output the ids of the messages in her list. In each line of your output, display the user id (not username) followed by a colon, and then the message ids separated by a single space. The program output must be written to a text file.

Assume that the input is correct syntactically and semantically (e.g., there wont be a message from an undefined user, etc.). Do not beautify your output. The grading of your output will be done by a program which will be expecting an input 100% complying with the specifications.

### Sample input:

```

ADD SH4RPSH00T3R11
ADD MARKHAMILL33
ADD IFollowAllTopics
ADD IFollowNoTopic
ADD IWillGetDeleted:(
ADD HowToUnsubscribe
SUBSCRIBE SH4RPSH00T3R11 Basketball
SUBSCRIBE MARKHAMILL33 StarWars
SUBSCRIBE IFollowAllTopics Basketball

```

SUBSCRIBE IFollowAllTopics StarWars  
SUBSCRIBE HowToUnsubscribe Basketball  
SEND IFollowNoTopic StarWars  
SEND IFollowAllTopics Basketball  
SEND IWillGetDeleted:( StarWars  
DELETEUSER IWillGetDeleted:(  
ADD YetAnotherUser  
UNSUBSCRIBE HowToUnsubscribe Basketball  
SUBSCRIBE HowToUnsubscribe StarWars  
SEND MARKHAMILL33 StarWars  
SEND MARKHAMILL33 Basketball

**Sample output:**

1: 2 5  
2: 1 4 5  
3: 1 2 4 5  
4: 1  
6: 4  
7: