CS 201, Fall 2023
Homework Assignment 3

Due: 23:59, December 6, 2023

In this homework, you will implement a simple social media platform. The simple social media platform consists of regular users with unique IDs, content creators with unique IDs, and contents with unique IDs throughout the system. Note that these ID types are independent of each other. For instance, there cannot be more than one regular user with ID 6, but there can be a regular user with ID 6 and a content creator with ID 6. They would be different persons. Regular users can follow content creators. For each content creator, you will store 2 sorted linked lists: one to store IDs of regular users following that content creator and one to store the contents shared by that content creator. In your implementation, you MUST use **sorted linked lists** for storing the regular users, content creators, and contents.

The simple social media platform will have the following functionalities. The details of these functionalities are given below:

1. Add a new content creator
2. Remove a content creator
3. Content creator adds a new content
4. Content creator removes a content
5. Add a regular user
6. Remove a regular user
7. A regular user follows a content creator
8. A regular user unfollows a content creator
9. Show the list of all regular users following a content creator (in sorted order according to their IDs)
10. Show the list of all contents of a content creator (in sorted order according to their IDs)
11. Show the list of all regular users (in sorted order according to their IDs)
12. Show the list of all content creators (in sorted order according to their IDs)

Add a regular user: The social media platform will allow users to sign up with an ID and a name. Since the regular user IDs must be unique, the system must check whether or not the specified regular user ID already exists, and if it exists, it must not allow the operation and display a warning message. Initially, a regular user is not following any content creator.
Example log messages:
Added regular user 6.
Cannot add regular user. There is already a regular user with ID 6.

Remove a regular user: The social media platform will allow removing regular user accounts. If the regular user does not exist, the system must display a warning message.
Example log messages:
Removed regular user 6.
Cannot remove regular user. There is no regular user with ID 6.

Add a new content creator: The social media platform will allow new content creators to sign up with an ID and a name. Since the content creator IDs must be unique, the system must check whether or not the specified content creator ID already exists, and if it exists, it must not allow the operation and display a warning message. Initially, a content creator is not followed by any users and does not have any content.
Example log messages:
Added content creator 6.
Cannot add content creator. There is already a content creator with ID 6.

Remove a content creator: The social media platform will allow removing content creator accounts. If the content creator does not exist, the system must display a warning message.
Example log messages:
Removed content creator 6.
Cannot remove content creator. There is no content creator with ID 6.

Content creator adds a new content: The social media platform will allow content creators to share new contents with a content ID and a title. Since the content IDs must be unique throughout the system, the system must check whether or not the specified content ID already exists, and if it exists, the system must not allow the operation and display a warning message. If the given content creator ID does not exist, the system must not allow the operation and display a warning message.
Example log messages:
Added content 6.
Cannot add content. There is already a content with ID 6.
Cannot add content. There is no content creator with ID 7.
Note: If the following two happen at the same time: 1) content ID already exists and 2) content creator does not exist, then the log message should be in the following format: Cannot add content. There is already a content with ID 6.

Content creator removes a content: Contents can be removed by specifying their content creator ID and content ID. If the content creator ID does not exist, the system must not allow the operation and display a warning message. If the content creator does not have a content with the given ID, the system must not allow the operation and display a warning message.
Example log messages:
Removed content 6.
Cannot remove content. There is no content creator with ID 7.
Cannot remove content. There is no content with ID 6 shared by content creator with ID 7.

A regular user follows a content creator: The social media platform will allow regular users to follow content creators. If the content creator ID and/or regular user ID does not exist, the system must display a warning message. If the user is already following the content creator, do not perform the operation and display a warning message.
Example log messages:
Regular user with ID 5 followed content creator with ID 6.
Cannot follow. Regular user and/or content creator ID does not exist.
Cannot follow. The user is already following the content creator.

A regular user unfollows a content creator: The social media platform will allow regular users to unfollow content creators. If the content creator ID and/or regular user ID does not exist, the system should not perform the operation and should display a warning message. If the user is not following the content creator, do not perform the operation and display a warning message.
Example log messages:
Regular user with ID 6 unfollowed content creator with ID 7.

Cannot unfollow. Regular user and/or content creator ID does not exist.
Cannot unfollow. The user is not following the content creator.

Show the list of all contents of a content creator (in sorted order according to their IDs): The social media platform will allow the display of a list of contents shared by a content creator in ASCENDING ORDER according to content IDs. If the content creator ID does not exist, the system should display a warning message.
Example log message 1:
Contents of content creator with ID 6:
1
6
Example log message 2:
Contents of content creator with ID 6:
None
Example log message 3:
Cannot show. There is no content creator with ID 6.

Show the list of all regular users following a content creator (in sorted order according to their IDs): The social media platform will allow the display of a list of regular users following a content creator in ASCENDING ORDER according to regular user IDs. If the content creator ID does not exist, the system should display a warning message. Example log message 1:
Regular users following content creator with ID 6:
1
6
Example log message 2:
Regular users following content creator with ID 6:
None
Example log message 3:
Cannot show. There is no content creator with ID 6.

Show the list of all regular users (in sorted order according to their IDs): The social media platform will allow the display of a list of all regular users' IDs and names. The entries should be in ASCENDING ORDER according to IDs.
Example log message 1:
Regular users in the social media platform:
None
Example log message 2:
Regular users in the social media platform:
1, Alper
4, Merve
6, Ayse

Show the list of all content creators (in sorted order according to their IDs): The social media platform will allow the display of a list of all content creators' IDs, names, number of followers and number of contents. The entries should be in ASCENDING ORDER according to IDs.
Example log message 1:
Content creators in the social media platform:
None
Example log message 2:
Content creators in the social media platform:
1, Gamer, 3 follower(s), 0 content(s)
3, Lonely Artist, 2 follower(s), 1 content(s)

Below is the required `public` part of the `SocialMediaPlatform` class that you must write in this assignment. The name of the class must be `SocialMediaPlatform`, and must include these public member functions. The interface for the class must be written in the file called `SocialMediaPlatform.h` and its implementation must be written in the file called `SocialMediaPlatform.cpp`. You can define additional public and private member functions and data members in this class. You can also define additional classes in your solution and implement them in separate files.

```
class SocialMediaPlatform {

public:
   SocialMediaPlatform();
   ˜SocialMediaPlatform();

   void addRegularUser( const int regularUserId, const string name );
   void removeRegularUser( const int regularUserId );

   void addContentCreator( const int contentCreatorId, const string name );
   void removeContentCreator( const int contentCreatorId );

   void addContent( const int contentCreatorId, const int contentId, const string
      title );
   void removeContent( const int contentCreatorId, const int contentId );

   void followContentCreator( const int regularUserId, const int contentCreatorId );
   void unfollowContentCreator( const int regularUserId, const int contentCreatorId);

   void showFollowersOf( const int contentCreatorId ) const;
   void showContentsOf( const int contentCreatorId ) const;

   void showAllRegularUsers( ) const;
   void showAllContentCreators( ) const;
};
```

Here is an example test program that uses this class and the corresponding output. We will use a similar program to test your solution so make sure that the name of the class is `SocialMediaPlatform`, its interface is in the file called `SocialMediaPlatform.h`, and the required functions are defined as shown above. Your implementation should use the format given in the example output to display the messages expected as the result of the defined functions.

Example test code:

```
#include <iostream>
using namespace std;

#include "SocialMediaPlatform.h"

int main() {

   SocialMediaPlatform SMP;

   SMP.showAllRegularUsers();
   cout << endl;
   SMP.showAllContentCreators();
   cout << endl;

   SMP.addContentCreator( 3, "Lonely Artist" );
   SMP.addContentCreator( 2, "Eater Man" );
```

```
    SMP.addContentCreator( 4, "Pet Lover" );
    SMP.removeContentCreator( 4 );
    SMP.removeContentCreator( 4 );
    SMP.addContentCreator( 4, "Musician Guy" );
    SMP.addContentCreator( 4, "Cyborg" );
    SMP.addContentCreator( 1, "Gamer" );
    SMP.addContentCreator( 6, "News" );
    cout << endl;

    SMP.addContent(3,1, "How to perform art?");
    SMP.addContent(4,2, "My favourite song");
    SMP.addContent(2,2, "Best Recipe Ever");
    SMP.addContent(6,5, "News of October");
    SMP.addContent(6,6, "News of November");
    SMP.addContent(6,8, "News of December");
    SMP.addContent(6,13, "News of 2023");
    SMP.removeContent(6, 5);
    SMP.removeContent(10, 7);
    SMP.removeContent(6, 7);

    SMP.showAllContentCreators();
    cout << endl;



    SMP.addRegularUser( 1, "Alper" );
    SMP.addRegularUser( 5, "Ali" );
    SMP.addRegularUser( 4, "Merve" );
    SMP.addRegularUser( 2, "Ferit" );
    SMP.addRegularUser( 2, "Burak" );
    SMP.addRegularUser( 6, "Ayse" );
    SMP.removeRegularUser( 2 );
    SMP.removeRegularUser( 7 );
    cout << endl;

    SMP.followContentCreator(1,6);
    SMP.followContentCreator(1,2);
    SMP.followContentCreator(1,1);
    SMP.followContentCreator(4,4);
    SMP.followContentCreator(4,3);
    SMP.followContentCreator(6,4);
    SMP.followContentCreator(6,6);
    SMP.followContentCreator(6,3);
    SMP.followContentCreator(5,3);
    SMP.followContentCreator(5,1);
    SMP.followContentCreator(1,7);
    SMP.followContentCreator(10,1);
    SMP.followContentCreator(10,10);
    SMP.unfollowContentCreator(1,7);
    SMP.unfollowContentCreator(10,1);
    SMP.unfollowContentCreator(10,10);
    SMP.unfollowContentCreator(1,3);
    SMP.unfollowContentCreator(6,4);
    cout << endl;

    SMP.removeRegularUser( 5 );
    SMP.removeContentCreator( 3 );
    cout<<endl;

    SMP.showAllContentCreators();
    cout << endl;
    SMP.showAllRegularUsers();
    cout << endl;
    SMP.showFollowersOf(6);
    cout << endl;
    SMP.showFollowersOf(10);
    cout << endl;
    SMP.showContentsOf(6);
    cout<<endl;
    SMP.showContentsOf(10);

    return 0;
}
```

Output of the example test code:

```
Regular users in the social media platform:
None

Content creators in the social media platform:
None

Added content creator 3.
Added content creator 2.
Added content creator 4.
Removed content creator 4.
Cannot remove content creator. There is no content creator with ID 4.
Added content creator 4.
Cannot add content creator. There is already a content creator with ID 4.
Added content creator 1.
Added content creator 6.

Added content 1.
Added content 2.
Cannot add content. There is already a content with ID 2.
Added content 5.
Added content 6.
Added content 8.
Added content 13.
Removed content 5.
Cannot remove content. There is no content creator with ID 10.
Cannot remove content. There is no content with ID 7 shared by content creator with ID
     6.
Content creators in the social media platform:
1, 0 follower(s), 0 content(s)
2, 0 follower(s), 0 content(s)
3, 0 follower(s), 1 content(s)
4, 0 follower(s), 1 content(s)
6, 0 follower(s), 3 content(s)

Added regular user 1.
Added regular user 5.
Added regular user 4.
Added regular user 2.
Cannot add regular user. There is already a regular user with ID 2.
Added regular user 6.
Removed regular user 2.
Cannot remove regular user. There is no regular user with ID 7.

Regular user with ID 1 followed content creator with ID 6.
Regular user with ID 1 followed content creator with ID 2.
Regular user with ID 1 followed content creator with ID 1.
Regular user with ID 4 followed content creator with ID 4.
Regular user with ID 4 followed content creator with ID 3.
Regular user with ID 6 followed content creator with ID 4.
Regular user with ID 6 followed content creator with ID 6.
Regular user with ID 6 followed content creator with ID 3.
Regular user with ID 5 followed content creator with ID 3.
Regular user with ID 5 followed content creator with ID 1.
Cannot follow. Regular user and/or content creator ID does not exist.
Cannot follow. Regular user and/or content creator ID does not exist.
Cannot follow. Regular user and/or content creator ID does not exist.
Cannot unfollow. Regular user and/or content creator ID does not exist.
Cannot unfollow. Regular user and/or content creator ID does not exist.
Cannot unfollow. Regular user and/or content creator ID does not exist.
Cannot unfollow. The user is not following the content creator.
Regular user with ID 6 unfollowed content creator with ID 4.

Removed regular user 5.
Removed content creator 3.

Content creators in the social media platform:
1, 1 follower(s), 0 content(s)
2, 1 follower(s), 0 content(s)
4, 1 follower(s), 1 content(s)
6, 2 follower(s), 3 content(s)

Regular users in the social media platform:
1, Alper
4, Merve
6, Ayse

Regular users following content creator with ID 6:
1
6

Cannot show. There is no content creator with ID 10.

Contents of content creator with ID 6:
6
8
13

Cannot show. There is no content creator with ID 10.
```

IMPORTANT NOTES:

Do not start your homework before reading these notes!!!

NOTES ABOUT IMPLEMENTATION:

1. You ARE NOT ALLOWED to modify the given parts of the header file. You MUST use sorted linked lists in your implementation. You will get no points if you use dynamic or fixed-sized arrays or any other data structures such as vectors/arrays from the standard library. However, if necessary, you may define additional data members and member functions. You may also define additional classes.
2. Moreover, you ARE NOT ALLOWED to use any global variables or any global functions.
3. Output message for each operation MUST match the format shown in the output of the example code.
4. Your code MUST NOT have any memory leaks. You will lose points if you have memory leaks in your program even though the outputs of the operations are correct. To detect memory leaks, you may want to use Valgrind which is available at `http://valgrind.org`.

NOTES ABOUT SUBMISSION:

1. In this assignment, you must have separate interface and implementation files (i.e., separate `.h` and `.cpp` files) for your class. Your class name MUST BE `SocialMediaPlatform` and your file names MUST BE `SocialMediaPlatform.h` and `SocialMediaPlatform.cpp`. Note that you may write additional class(es) in your solution.
2. The code (`main` function) given above is just an example. We will test your implementation using different scenarios, which will contain different function calls. Thus, do not test your implementation only by using this example code. We recommend you to write your own driver files to make extra tests. However, you MUST NOT submit these test codes (we will use our own test code). In other words, do not submit a file that contains a function called `main`.
3. You should put all of your `.h` and `.cpp` files into a folder and zip the folder (in this zip file, there should not be any file containing a `main` function). The name of this zip file should conform to the following name convention: secX-Firstname-Lastname-StudentID.zip where X is your section number. The submissions that do not obey these rules will not be graded.
4. Make sure that each file that you submit (each and every file in the archive) contains your name, section, and student number at the top as comments.
5. You are free to write your programs in any environment (you may use Linux, Windows, MacOS, etc.). On the other hand, we will test your programs on "dijkstra.ug.bcc.bilkent.edu.tr" and we will expect your programs to compile and run on the dijkstra machine. If we could not get your program properly work on the dijkstra machine, you would lose a considerable amount of points. Thus, we recommend you to make sure that your program compiles and properly works on dijkstra.ug.bcc.bilkent.edu.tr before submitting your assignment.
6. This assignment is due by 23:59 on Sunday, December 6, 2023. You should upload your work to Moodle before the deadline. No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course home page for further discussion of the late homework policy.
7. We use an automated tool as well as manual inspection to check your submissions against plagiarism. Please see the course home page for further discussion of academic integrity and the honor code for programming courses in our department.
8. This homework will be graded by your TA **Burak Ferit Aktan** (ferit.aktan@bilkent.edu.tr). Thus, you may ask your homework related questions directly to him. There will also be a forum on Moodle for questions.