/\* In this exercise we want to design a simple social network.

\* We have users which have a name, an age, a height (in cm), a set of hobbies,

\* an entry whether they are male or female, an internal ID and a set of friends.

\* A user doesn't have to provide all information except the name. The name always has to be provided. (think about how to design this)

\*

\* Our SocialNetwork should provide us with the possibility to add and delete users

\* from the network. It should also allow us to search users by name, age, id or hobbies.

\* Eg.:

\* searchUserByName("Jochen");

\* searchUserByAge(36);

\* searchUserByHobbies({"Jogging", "Football", "Tennis"}); //Returns alls users which have the hobbies jogging, footbal AND tennis.

\* Additionally we want to receive all the friends a particular user has. But we only want to receive the set of ID's of those users (friends).

... getFriendsOfUser(...)

\*

\* Neither the names of the users have to be unique (e.g. we can have multiple Klaus in the network) nor any other attribute.

\* Only the id of every user has to be unique in the whole network.

\*

\* The skeleton code below can be used as a starting point but doesn't have to.

\* The comments "fill in" are placeholders where you definitely have to put in some code when

\* you use this skeleton code. But this doesn't mean that you shouldn't or can't put code anywhere else.

\*

\* Also write some simple unit tests to show how you would test the functionality of the Network.

\* Don't use any testing framework. Simple if-statements are sufficient for this exercise.

\*

\* Hint: Think about performance versus memory tradeoffs in your design, so you can give good

\* reasons for your decision.

\*/

class User

{

/\*fill in\*/

};

class SocialNetwork

{

public:

void addUser(/\*fill in\*/);

void deleteUser(/\*fill in\*/);

/\*fill in\*/ searchUserByName(/\*fill in\*/);

/\*fill in\*/ searchUserByAge(/\*fill in\*/);

/\*fill in\*/ searchUserByHobbies(/\*fill in\*/);

/\*fill in\*/ getFriendsOfUser(/\*fill in\*/);

private:

/\*fill in\*/

};