**Access Control System**

There is an Access Control System, which grants or denies access in response to a specified access point identifier and user identifier. At the beginning the system is unconfigured, it has no registered access points and no registered users. You can create and delete access points and users by providing numerical identifiers. The numerical identifiers (both for access points and users) must be in the integer range of 1 to 100 and must be unique. You can also grant and revoke access to a certain access point and check whether the user has access to an access point. In case of wrong arguments, an error message appears and the execution is interrupted.

**How to use the program:**

A list of input sequences have to be given as command line argument. You can use the following commands:

* CreateAP: creates a new access point, you have to provide a new unique access point identifier. Output: DONE/FAIL
* CreateUser: creates a new user, you have to provide a new unique user identifier. Output: DONE/FAIL
* DeleteAP: deletes the access point, you have to provide an existing access point identifier. Output: DONE/FAIL
* DeleteUser: deletes the user, you have to provide an existing user identifier. Output: DONE/FAIL
* Grant: grants access to an access point, you have to provide the access point identifier, then the user identifier. Output: DONE/FAIL
* Revoke: revokes access to an access point from a user, you have to provide the access point identifier first then the user identifier. Output: DONE/FAIL
* Check: checks whether the user has access to an access point, you have to provide the access point identifier first then the user identifier. Output: YES/NO

Example:

./accessControlSystem [[createAP, 1], [createAP, 1], [createUser, 1], [createUser, 1], [grant, 1, 1], [grant, 1, 1], [grant, 2, 2], [check, 1, 1], [revoke, 1, 1], [revoke, 1, 1], [revoke, 2, 2], [check, 1, 1], [check, 2, 2], [deleteap, 1], [deleteap, 1], [deleteuser, 1], [deleteuser, 1]]

* createap 1 // creates an access point
* createap 1 // already exists, can’t be created again
* createuser 1 // create user
* createuser 1 //user already exists, can’t be created again
* grant 1 1 // grant access
* grant 1 1 // the access already exist, can’t be granted again
* grant 2 2 // can’t grant the access, neither the access point, nor the user exist
* check 1 1 // there exist access to AP 1 by user 1
* revoke 1 1 // revokes the access
* revoke 1 1 // nothing to revoke
* revoke 2 2 // can’t revoke access, neither the access point, nor the user exist
* check 1 1 // there is no access to AP 1 by user 1
* check 2 2 // neither the access point, nor the user exist
* deleteAP 1 // successfully deleted AP 1
* deleteAP 1 // can't be deleted, AP 1 doesn’t exist
* deleteUser 1 // successfully deleted user 1
* deleteUser 1 // can't be deleted, user 1 doesn’t exist

Output: The output is a list of the results of the commands. If a command was successful, its output is yes, otherwise no.

[DONE, FAIL, DONE, FAIL, DONE, FAIL, FAIL, YES, DONE, FAIL, FAIL, NO, NO, DONE, FAIL, DONE, FAIL]