**Friendster CLI**

Hi, PPTI 16 fellows. In purpose to practice on how you must think when creating a program in a computer using array and string manipulations, this practice case will help you to translate from algorithm and procedures to source code in C programming language. Hope this helps!

In this case we wan you to make a program that can add friends.

When the program has been run, here’s the program flow that you need to create:

* Before showing main menu, **insert** these **data** to the program as list of users. (Feel free to insert it by reading file or manually insert to array)

|  |  |  |
| --- | --- | --- |
| **User ID** | **Name** | **Email** |
| christ0208 | Christopher | [chris@gmail.com](mailto:chris@gmail.com) |
| jordywa | Jordy | [jordy@gmail.com](mailto:jordy@gmail.com) |
| haku | Haku | [haku@gmail.com](mailto:haku@gmail.com) |
| elior | Eleanor | [el@gmail.com](mailto:el@gmail.com) |
| cleo | Cleo Deus | [cleo@gmail.com](mailto:cleo@gmail.com) |

* In the beginning of program’s execution, it will show **three menu options**.
  + Add new Friend
  + Unfriend
  + Exit

Background pattern

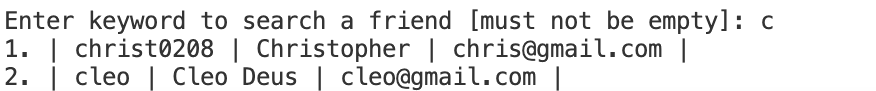
Description automatically generated with low confidence

**Figure 1 Main menu**

* If user inputs **Menu 1** (Add New Friend), do:
  + - **Ask** user to input **keyword**,that hints to specific name of the user or leading to specific user ID
    - Then, **show** users that **keyword** **equals** to specific user ID (**case insensitive**) or **keyword exists** in specific user name (**case insensitive**).

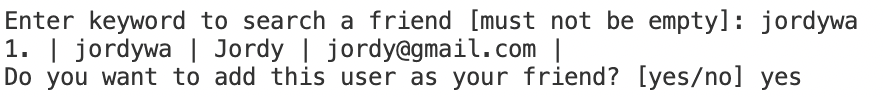


**Figure 2 Add new friend (search by user ID)**

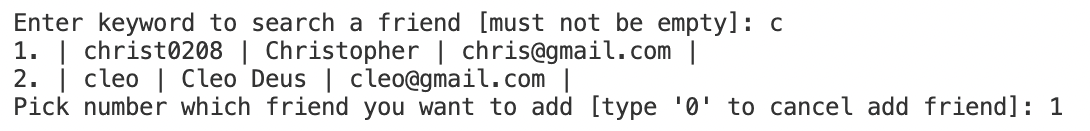


**Figure 3 Add new friend (search by user name)**

* + - If **users** with those conditions are **not found**, **return** to main menu.
    - Otherwise, if **only 1 user** qualified with those conditions, **prompt confirmation** whether the selected user want to be **added as a friend**.
    - Otherwise, **pick 1 user** using number from listed users with qualified conditions above. Then, **add** picked user to be added as a **friend**.



**Figure 4 Confirmation when only 1 user is found**



**Figure 5 User selection when more than 1 user is found**

* If user input **Menu 2** (Unfriend), do:
  + **Print** **list of friends** that you have added.
  + **Ask** user to pick a friend that want to be unfriended using **index number**.
  + **Remove** the selected user from friend list.

Text

Description automatically generated

**Figure 6 Unfriend menu**

* If user input **Menu 3** (Exit), **close** the **application**.