

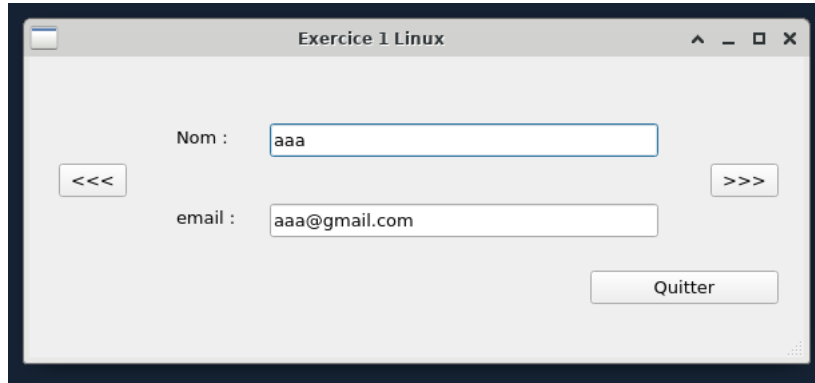
Exercice n°1 : le makefile

Objectifs :

- Se familiariser avec l'utilitaire **make** et la création d'un fichier **makefile**
- Se familiariser avec l'utilisation de la librairie graphique Qt

Description générale :

L'application ressemble visuellement à



Elle permet d'afficher le nom et l'email des individus gérés par l'application.

La fenêtre graphique a été créée pour vous et vous est fournie sous la forme d'un ensemble de fichiers sources.

Etape 1 : Création d'un fichier makefile

On vous fournit une archive **UNIX_Exercice1.tar** qui contient les fichiers suivants :

- **main.cpp** : le main de votre application
- **mainwindow.cpp** : code source de la fenêtre de l'application, seul fichier que vous devez modifier.
- **mainwindow.h** : fichier header correspondant au fichier mainwindow.cpp.
- **moc_mainwindow.cpp** et **ui_mainwindow.h** : fichiers sources propres à Qt nécessaires à la fenêtre graphique.
- **Compile.sh** : fichiers texte (on appelle ça un script) contenant les lignes de compilation nécessaires pour la création du fichier exécutable « UNIX_Exercice1 ».

Il est déjà possible de compiler l'application en utilisant le fichier **Compile.sh** :

```
# sh Compile.sh
# UNIX_Exercice1
```

Le but est donc de créer un fichier **makefile** permettant de réaliser la compilation de tous les fichiers nécessaires à la création de l'exécutable UNIX_Exercice1.

Etape 2 : Programmation des fonctionnalités

Vous trouverez dans le fichier **mainwindow.cpp** :

```
typedef struct
{
    int    id;
    char   nom[40];
    char   email[40];
} ELEMENT;

ELEMENT Elm[] =
{
    {0, "", ""},
    {1, "aaa", "aaa@gmail.com"},
    {2, "bbb", "bbb@gmail.com"},
    {3, "ccc", "ccc@gmail.com"},
    {4, "ddd", "ddd@gmail.com"},
    {5, "eee", "eee@gmail.com"},
    {0, "", ""}
};
```

Le vecteur **Elm** contient l'ensemble des individus gérés par l'application. Aucun ajout/modification/suppression ne sera géré ici.

On vous demande à présent de modifier le code fourni (**uniquement dans le fichier mainwindow.cpp**) de telle sorte que :

- Un clic sur le bouton « **Quitter** » termine l'application → vous devez modifier la fonction
void MainWindow::on_pushButtonQuitter_clicked()
- Un clic sur le bouton « **>>>** » permet d'afficher l'élément suivant du vecteur **Elm** → vous devez modifier la fonction
void MainWindow::on_pushButtonSuivant_clicked()
- Un clic sur le bouton « **<<<** » permet d'afficher l'élément précédent du vecteur **Elm** → vous devez modifier la fonction
void MainWindow::on_pushButtonPrecedent_clicked().

N'oubliez pas de gérer le fait que l'on peut atteindre les « extrémités » du vecteur **Elm** et qu'il ne faut pas en sortir (on n'affiche pas les éléments d'id 0).

Afin de modifier le texte des champs de texte apparaissant dans la fenêtre (nom et email), vous disposez des méthodes

- **void MainWindow::setNom(const char* Text)**
- **void MainWindow::setEmail(const char* Text)**

que vous devez utiliser et non modifier.