Devoir surveillé

8 Novembre 2022 — Durée 1h15

Document autorisé : **Mémento C** vierge de toute annotation

On s'intéresse à une fonction de recherche dans un *annuaire* : un annuaire est une séquence de *personnes*, ordonnée par noms. Une *personne* est une structure composé d'un nom, d'un prénom et d'un numéro de téléphone (représenté ici par un simple entier).

Les spécifications des paquetages **personne**, **annuaire**, **es_personne**, **es_annuaire** et **recherche** sont fournies en annexe. Certaines lignes des fichiers d'en-tête sont volontairement masquées, elles ne sont pas utiles dans ce sujet. Pour les exercices de ce sujet, vous n'avez pas besoin de l'implémentation de ces paquetages ; il ne vous est pas demandé non plus de les réaliser.

Pour la fonction `lire_annuaire`, un annuaire dans un fichier doit respecter le format suivant :

- le fichier contient une liste de personnes, triée par ordre croissant de nom, puis de prénom entre deux personnes de noms identiques ;
- on suppose qu'il n'existe pas dans un annuaire deux personnes de mêmes noms *et* prénoms ;
- chaque ligne du fichier contient une personne au format : nom prénom numéro. Par facilité, on suppose les numéros de téléphone représentés par des entiers.

Par exemple :

```
1 Amidala Padmé 0472398092
2 Kenobi Obiwan 0687343529
3 Organa Leia 0660848356
4 Skywalker Anakin 0622245612
5 Skywalker Luke 0147200001
```

Exercice 1. (5 pt) Écrire un programme `recherche_annuaire.c` qui prend en argument de la ligne de commande un nom de fichier contenant un annuaire, et qui :

- lit cet annuaire
- lit sur l'entrée standard un nom et un prénom
- recherche la personne correspondant à ce nom et prénom dans l'annuaire lu
- si cette personne existe, affiche son numéro de téléphone ; affiche « Personne non trouvée » sinon.

Exercice 2. (5 pt)

1. Schématiser la structure globale et les relations entre les différents paquetages et le programme écrit à l'exercice précédent.
2. Écrire un Makefile permettant de compiler le programme `recherche_annuaire`. L'exécution de la commande `make` sans argument doit produire un exécutable nommé `recherche_annuaire`.

Exercice 3. (4 pt) Décrire un jeu de tests fonctionnels permettant de tester la fonction `recherche`, à l'aide du programme écrit à l'exercice 1.

NB : il est demandé de *décrire* ce jeu de tests, et donc de *justifier* sa construction, sans nécessairement écrire explicitement les valeurs de chaque test.

Exercice 4. (6 pt) Nous souhaitons utiliser un oracle, permettant d'automatiser le test de la fonction `recherche`. On se place spécifiquement dans le cas où la personne recherchée existe dans l'annuaire.

1. Dans ce cas, quelle propriété peut-on vérifier sur la valeur renvoyée par la fonction `recherche` ?
2. Écrire une fonction `oracle` qui permet de faire cette vérification.
3. Compléter le programme de l'exercice 1 pour utiliser cette fonction `oracle` : le programme doit afficher un message indiquant si la propriété indiquée est vraie ou non.

Pour cet exercice, on peut utiliser la fonction `strcmp` dont la spécification est donnée en annexe.

Annexes

Paquetage personne : contenu du fichier personne.h

```c
typedef struct {
    ...
} Personne;

// Renvoie le nom de la personne p
char * nom_personne(Personne * p);

// Renvoie le prénom de la personne p
char * prenom_personne(Personne * p);

// Renvoie le numéro de téléphone de la personne p
int num_telephone_personne(Personne * p);
```

Paquetage es_personne : contenu du fichier es_personne.h

```c
#include "personne.h"
#include <stdio.h>

/* Lit une personne p depuis le fichier f.
Préconditions : p est un pointeur vers une structure de type Personne
déjà allouée dans la mémoire, f est un fichier ouvert en lecture */
void lire_personne(FILE * f, Personne * p);

/* Ecrit une personne p dans un fichier f
Préconditions : p est un pointeur vers une structure de type Personne
déjà allouée dans la mémoire, f est un fichier ouvert en écriture */
void ecrire_personne(FILE * f, Personne *p);
```

Paquetage annuaire : contenu du fichier annuaire.h

```c
#include "personne.h"

typedef struct cellule {
    ...
} Cellule;

typedef Cellule * Annuaire;
```

Paquetage es_annuaire : contenu du fichier es_annuaire.h

```c
#include "annuaire.h"
#include "personne.h"
#include "es_personne.h"
#include <stdio.h>

/* Renvoie un annuaire lu depuis le fichier f.
Préconditions : f est un fichier ouvert en lecture, contient une liste de personnes au format :
- liste triée par nom, puis par prénom croissant
- une personne par ligne : nom prénom numéro */
Annuaire * lire_annuaire(FILE * f);

/* Ecrit un annuaire a dans un fichier f
Préconditions : f est un fichier ouvert en écriture */
void ecrire_annuaire(FILE * f, Annuaire * a);
```

Paquetage `recherche` : contenu du fichier `recherche.h`

```c
#include "personne.h"
#include "annuaire.h"

/* Recherche la personne de nom <nom> et de prénom <prenom> dans l'annuaire <ann>.
Si la personne existe dans l'annuaire, un pointeur vers la structure représentant
cette personne est renvoyée. Sinon, la valeur NULL est renvoyée.
*/
Personne * recherche(char * nom, char * prenom, Annuaire * ann);
```

Spécification de la fonction `strcmp`

```c
int strcmp(const char *s1, const char *s2);
```

La fonction `strcmp()` compare les deux chaînes `s1` et `s2`.

`strcmp()` retourne un entier indiquant le résultat de la comparaison comme suit :

- 0 si `s1` et `s2` sont égales ;
- une valeur négative si `s1` est inférieure à `s2` ;
- une valeur positive si `s1` est supérieure à `s2`.