R1.01 Initiation au développement – SAÉ2

11 janvier 2026
Erwan Kerrien

Naissances multiples

Rappel : si vous avez des questions sur cette SAÉ ou sur le cours, n'hésitez pas à m'envoyer un mail à Erwan.Kerrien@inria.fr

1 Consignes pour le rendu

Cette SAÉ est à faire en binôme.

Il est attendu un code source déposé sur l'instance gitlab de l'Université de Lorraine, accompagné d'un fichier `Makefile` pour compiler le projet et un fichier README.md qui le décrit (texte de présentation de votre projet sur gitlab). Vous devrez également ajouter un rapport. Le projet est à rendre pour le 11 janvier 2025 à 23h59 : un clone de votre projet sera fait automatiquement à précisément minuit le 12 janvier 2026.

Dès le début du projet, vous devez m'envoyer par mail l'url de clonage de votre projet, et ne pas oublier de me mettre `maintainer`. **Ces trois étapes (création du projet, envoi de l'url et ajout comme maintainer) doivent être faites pour le 16 décembre au plus tard.**

Éléments de notation :

- Le code source devra être correctement commenté.
- Votre projet doit compiler sans erreur ni avertissement grâce au `Makefile` fourni.
- Une fois compilé, votre programme doit pouvoir s'exécuter, en prenant en argument le nom du fichier CSV à analyser (pas de nom en dur dans le code source !)
- Chaque fonctionnalité demandée dans l'énoncé sera notée, la note étant plus douce si le rapport mentionne et explique les difficultés rencontrées, et la note étant plus sévère en cas d'erreur ou de plantage.
- Le programme devra notamment fonctionner même si le fichier CSV passé en entrée disparaît lors de son exécution. Autrement dit, toutes les données utiles qu'il contient doivent être stockées en mémoire.
- Le rapport sera rendu au format pdf. Il fera au minimum 2 pages, mais n'excédera pas 5 pages. Il devra discuter et justifier du choix des structures de données utilisées à la fois pour une information atomique (un élément), comme pour l'ensemble des informations à garder en mémoire. Il devra par ailleurs discuter des fonctionnalités et limitations du programme, en vous basant sur un ensemble de tests que vous décrirez. Les fichiers nécessaires aux tests seront ajoutés à l'archive.
- Le rapport devra indiquer précisément comment le travail a été partagé, et clairement identifier la part de chacun dans chaque tâche.

Très important : le dépôt gitlab ne devra pas contenir de fichier exécutable, ni de fichier objet (`*.o`), ni le fichier CSV qui doit être traité dans ce projet (il est gros et je l'ai récupéré par ailleurs). Vous pourrez cependant intégrer de **petits** fichiers CSV qui vous ont permis de faire des tests que vous jugerez pertinents. Les seules extensions autorisées pour les fichiers sont donc : `.c, .h, .pdf, .csv`, sachant que les fichiers CSV devront être petits. Vous devrez y ajouter un fichier `Makefile`. **Tout manquement à cette règle sera sanctionné.**

2 Objectifs de la SAÉ

L'INSEE a pour mission de collecter et analyser des informatiques statistiques sur l'économie et la société françaises. C'est en particulier le cas pour les prénoms des enfants nés en France entre 1900 et 2024. L'INSEE a regroupé dans un fichier, pour chaque prénom, chaque année, et chaque département, le nombre d'enfants concernés (nés cette année-là dans le département, avec ce prénom).

Toutes les informations sont accessibles depuis le site `https://www.insee.fr/fr/statistiques/8595130` (bien prendre le fichier par département, et non le fichier national). En particulier :

— le fichier est disponible depuis `https://www.insee.fr/fr/statistiques/fichier/8595130/prenoms-2024-dpt_csv.zip`
— la description du fichier et ses champs est donnée sous l'onglet "Dictionnaire" `https://www.insee.fr/fr/statistiques/8595130?sommaire=8595113#dictionnaire`

L'objectif de la SAÉ est de charger cette base de données, en la simplifiant éventuellement, afin de pouvoir l'interroger.
Le programme que vous concevrez devra proposer un prompt à l'utilisateur qui affichera :
`Que voulez-vous afficher ? (0 pour le menu) >`

En réponse, l'utilisateur devra entrer un chiffre qui permettra de lancer un affichage, selon les items suivants :

— 0 : Ce menu
— 1 : Le nombre de naissances
— 2 : Le nombre de prénoms
— 3 : Statistiques sur un prénom
— 4 : Quitter

L'item 0 : Ce menu. Affiche le menu.

L'item 1 : Le nombre de naissances. Affiche le nombre de naissances depuis 1900, à savoir le nombre total d'individus pris en compte dans ce fichier.

L'item 2 : Le nombre de prénoms. Commence par demander si on veut distinguer le genre ou pas, puis affiche le résultat. (par exemple Camille est un prénom à la fois masculin et féminin. Il comptera donc pour 2 prénoms si on distingue le genre, 1 seul sinon)

L'item 3 : Statistiques sur un prénom. Commence par demander le prénom. Puis, si le prénom est présent dans les deux genres, demande si on veut distinguer les genres. Affiche ensuite pour le prénom (et le genre éventuel) les informations suivantes :

— Nombre d'individus au total
— Année de première apparition
— Année de dernière apparition

L'item 4 : Quitter. Quitte le programme.

Voici un exemple de session (les nombres sont fantaisistes).

```
Que voulez-vous afficher ? (0 pour le menu) > 0
0: Ce menu
1: Le nombre de naissances
2: Le nombre de prénoms
3: Statistiques sur un prénom
4: Quitter
Que voulez-vous afficher ? (0 pour le menu) > 3
Quel prénom ? Roger
Le prénom Roger a été donné à 654 enfants.
Année de première apparition 1900.
Année de dernière apparition 1983.
Que voulez-vous afficher ? (0 pour le menu) > 2
Souhaitez-vous distinger le genre (O/N) > O
Le fichier recouvre 65122 prénoms masculins et 65328 prénoms féminins.
Que voulez-vous afficher ? (0 pour le menu) > 3
Quel prénom ? Camille
Distinguer le genre ? (O/N) O
Le prénom Camille a été donné à 1321 garçons et 1432 filles.
Année de première apparition 1904.
Année de dernière apparition 2020.
```

```
Que voulez-vous afficher ? (0 pour le menu) > 1
Le fichier recouvre 3789411 naissances.
Que voulez-vous afficher ? (0 pour le menu) > 2
Souhaitez-vous distinger le genre ? (O/N) N
Le fichier recouvre 14879 prénoms
Que voulez-vous afficher ? (0 pour le menu) > 0
0: Ce menu
1: Le nombre de naissances
2: Le nombre de prénoms
3: Statistiques sur un prénom
4: Quitter
Que voulez-vous afficher ? (0 pour le menu) > 4
```

3 Le format CSV

Le gros fichier que vous devez traiter est au format CSV (*Comma Separated Values*). Ce format est une manière très courante de stocker une base de données simple dans un seul fichier texte. Simple signifie que les informations sont tabulées : chaque information est un enregistrement qui contient un nombre prédéfini de champs. Les fichiers de ce type peuvent être par exemple importés dans un tableur (libreoffice calc ou excel pour les windowsiens) puisque chaque enregistrement peut être stocké dans une ligne, et chaque champ correspond à une colonne.

Dans le format CSV, chaque ligne du fichier correspond à un enregistrement, hormis la première ligne qui est spéciale et indique l'en-tête des colonnes (le nom de chaque champ). Dans une ligne, les champs sont séparés par un caractère : le fichier considéré ici utilise le caractère ; et le code demandé ne gérera que ce cas. Un champ est stocké sous forme de chaîne de caractères. Un champ peut être vide.

Le fichier suivant, appelé `Test.csv`, est un exemple de fichier CSV avec ; pour séparateur, comportant 9 enregistrements, chaque enregistrement comportant quatre champs : Nom, Prénom, Âge et Genre. On remarque que le fichier contient 10 lignes : la première ligne indique le nom de chaque champ.

```
Nom;Prénom;Âge;Genre
Enfant;Hélène;44;F
Enfant;Ludivine;47;F
Flaille;Abdel;19;M
Flaille;Akim;23;M
Flaille;Yves;21;M
Neymar;Jean;24;M
Titegoute;Justine;78;F
Yapudebiairedenlefrigo;Robin;67;M
Kerrien;Erwan;;M
```

Attention : le fichier ci-dessus n'a absolument pas les mêmes champs que le fichier cible de cette SAÉ.

4 Pistes et suggestions

Les éléments suivants ne sont donnés qu'à titre de suggestions et ne comportent aucun caractère obligatoire. Il est rappelé cependant que le rendu final devra comprendre un rapport justifiant les choix effectués et décrivant les tests essentiels réalisés. Cependant, dans les sections qui suivent, des fonctions et outils proposés par la librairie standard de C vous seront présentées, et peut-être y trouverez-vous une quelconque utilité.

En premier lieu, vous pourrez utiliser les booléens définis par le type `bool` dans `stdbool.h`. Ce fichier définit entre autres les constantes `true` et `false`. On l'emploie en ajoutant la directive `#include <stdbool.h>` en début de fichier source C.

Exercice : Lecture de fichier CSV

Un bon exercice préparatoire consiste à programmer un lecteur de fichier CSV. Le programme pourra par exemple prendre en paramètre un fichier et afficher successivement les enregistrements qu'il contient. Par exemple la command `readCSV`

`Test.csv` (voir `Test.csv` ci-dessus) affichera

```
New line:
[0] 'Enfant' [1] 'Hélène' [2] '44' [3] 'F
'
New line:
[0] 'Enfant' [1] 'Ludivine' [2] '47' [3] 'F
'
New line:
[0] 'Flaille' [1] 'Abdel' [2] '19' [3] 'M
'
New line:
[0] 'Flaille' [1] 'Akim' [2] '23' [3] 'M
'
New line:
[0] 'Flaille' [1] 'Yves' [2] '21' [3] 'M
'
New line:
[0] 'Neymar' [1] 'Jean' [2] '24' [3] 'M
'
New line:
[0] 'Titegoute' [1] 'Justine' [2] '78' [3] 'F
'
New line:
[0] 'Yapudebiairedenlefrigo' [1] 'Robin' [2] '67' [3] 'M
'
New line:
[0] 'Kerrien' [1] 'Erwan' [2] '' [3] 'M
'
```

Plutôt que d'utiliser `fscanf`, vous pourrez utiliser à profit les trois fonctions suivantes disponibles en insérant les directives `#include <stdio.h>` et `#include <string.h>` en début de fichier source (dont vous pouvez accéder à l'aide via la commande `man <nom de onction>`) : `getline`, `strsep` et `feof`.

Bonus : vous pouvez remarquer que le genre se termine par un retour à la ligne (l'apostrophe ' est à la ligne au lieu de suivre directement la lettre de genre). Modifiez votre programme pour qu'il ne s'affiche pas.

Exercice : Structures de données

Une première question à se poser est celle des informations qui sont intéressantes dans un enregistrement. Doit-on tout conserver ou pas ? Quelle structure de données, qu'on pourrait appeler élémentaire, serait intéressante pour stocker l'information intéressante ? La lecture attentive de la description du fichier CSV disponible sur le site de l'INSEE vous aidera pour le choix des types pour encoder chaque information pertinente.

Au-delà, quelle structure de données serait pertinente pour mettre ensemble toutes ces structures élémentaires ? Le critère principal ici sera la performance de l'accès à l'information demandée par l'utilisateur, en termes de temps de calcul, Il faut notamment exclure une relecture du fichier à chaque requête et donc tout charger en mémoire dès le lancement du programme.

Dans ce travail, il ne faudra pas hésiter à se tromper. C'est l'occasion de tester différentes options, et d'imaginer les tests qui permettent de sélectionner la plus efficace. Pour cela cependant, il faudra bien concevoir votre architecture de code afin de ne pas avoir à tout réécrire et minimiser le volume de code à écrire pour chaque option.