

# Examen

## Langage Procédural : C

15 mars 2022

*La compréhension du sujet fait partie de l'examen. Il ne sera donc répondu à aucune question.  
Le barème prend en compte l'argumentation fournie dans les commentaires associés aux réponses.*

*Tout document et fichier source personnel autorisés, documentation du langage C autorisée, communications électroniques interdites. Les bibliothèques standard du C sont autorisées.*

*Durée 1h30 individuel ; barème approximatif*

Vous devez créer dans la fonction `main` le code vous ayant permis de tester les fonctions répondant à chaque question. Vous devez fournir vos sources et script de compilation (ou Makefile), sans les exécutables, que vous posterez sous la forme d'une seule archive sur *madoc.univ-nantes.fr* dans l'emplacement prévu dans le cours *INFO3 - S6 - Langage C*.

Veillez à respecter la séparation de la déclaration du prototype et du corps de vos fonctions. Il est **TRÈS** fortement recommandé de compiler vos programmes régulièrement au fil de votre avancement dans les questions.

## Annuaire simplifié

Nous désirons créer un programme gérant en mémoire un répertoire d'étudiants. Cet annuaire devra pouvoir être interrogé avec des index différents.

*Pour chaque question, pensez bien à tester votre code dans la fonction `main`.*

*La nature et le nombre d'argument de chaque fonction à créer est à votre discrétion et fait partie de la notation.*

*Les questions peuvent être traitées de manière indépendantes et être testées en fournissant un jeu de données écrit directement dans le code.*

*Vous pouvez factoriser du code en créant des fonctions qui ne sont pas demandées explicitement dans les différentes questions.*

### QUESTION 1. (2 points) STRUCTURE DE DONNÉE ÉTUDIANT

Créer une structure mémoire permettant de stocker les informations associées à un étudiant : une chaîne de caractères pour le *nom*, une chaîne de caractères pour le *prénom*, un entier pour l'*âge*, un entier pour l'*année d'inscription*. Cette structure ne devrait pas être connue du développeur utilisateur de votre annuaire.

### QUESTION 2. (2 points) STRUCTURE DE DONNÉE ANNUAIRE

Créer une structure mémoire permettant de stocker les informations associées à un annuaire. Ce dernier doit contenir les données des étudiants ainsi que les index (à venir dans une prochaine question), et sera le point d'entrée de toutes les fonctions manipulant votre annuaire.

### QUESTION 3. (2 points) CRÉATION

Créer une fonction `GenererAnnuaire` permettant de créer un nouvel annuaire vide. Cette fonction ne prends pas de paramètre et doit retourner l'annuaire créée.

### QUESTION 4. (2 points) DESTRUCTION

Créer une fonction `SupprimerAnnuaire` permettant de supprimer correctement toutes les données en mémoire stockées dans l'annuaire passé en paramètre.

QUESTION 5. (2 points) SAISIE UTILISATEUR

Créer une fonction **SaisirAnnuaire** permettant d'ajouter plusieurs entrées dans votre annuaire. Cette fonction prends donc en paramètre l'annuaire à compléter et demande à l'utilisateur de saisir au clavier les données à ajouter.

Pour chaque entrée, vous devrez demander à l'utilisateur de remplir tous les champs *nom*, *prénom*, *âge* et *année d'inscription*. L'utilisateur doit pouvoir saisir plusieurs utilisateurs. Si des données sont déjà présentes dans l'annuaire, celles-ci doivent rester non modifiées. La recherche de doublon n'est pas à considérer ici.

Pour cet exercice, la saisie ne doit pas être robuste, et vous pouvez vous contenter de l'utilisation de **scanf**.

QUESTION 6. (2 points) AFFICHAGE

Créer une fonction **AfficheAnnuaire** permettant d'afficher sur la console les étudiants stockés dans un annuaire passé en paramètre.

QUESTION 7. (2 points) ENREGISTREMENT DANS UN FICHIER

Créer une fonction **EnregistrerAnnuaire** permettant de sauvegarder les étudiants stockés dans un annuaire passé en paramètre. Le nom du fichier sera également passé en paramètre. Le format du fichier est à votre discrétion.

QUESTION 8. (2 points) LECTURE DEPUIS UN FICHIER

Créer une fonction **LireAnnuaire** prenant en paramètre un nom de fichier. Cette fonction crée un nouvel annuaire et l'initialise avec les données contenues dans le fichier (créé précédemment par la fonction **EnregistrerAnnuaire**).

QUESTION 9. (2 points) INDEXATION

Dans cette question, nous cherchons à créer des index qui faciliteront la recherche d'un étudiant dans l'annuaire. L'idée est de créer des tableaux de pointeurs vers les données stockées dans l'annuaire et de les trier afin de pouvoir utiliser un algorithme de recherche dichotomique.

Ajouter une fonction **IndexerAnnuaire** (non accessible directement au développeur utilisateur de votre annuaire) créant 3 index : un relatif aux noms des étudiants, un relatif aux prénoms des étudiants et un dernier relatif à l'âge des étudiants. Ces index doivent être stockés dans l'annuaire. Vous devrez appeler cette fonction dans chaque fonction précédemment créée qui modifie le contenu de l'annuaire. Pour cette question, vous devrez utiliser une fonction de tri. Vous pouvez soit prendre la fonction **quicksort** que vous avez développés en TP, soit la fonction **qsort** de la bibliothèque standard.

QUESTION 10. (2 points) RECHERCHE

Créer 3 fonctions de recherche renvoyant sur la sortie standard les entrées d'un annuaire correspondant au critère choisi :

- une fonction recherchant des entrées relatives à un nom d'étudiant
- une fonction recherchant des entrées relatives à un prénom d'étudiant
- une fonction recherchant des entrées relatives à un âge d'étudiant