

NOM:
PRÉNOM:
GROUPE:

Programmation C

15 janvier 2015

Documents non autorisés.

Durée: 1h30

Vous pouvez omettre les directives `#include` dans vos réponses. Par ailleurs, vous apporterez un très grand soin à la présentation car elle interviendra dans la notation.

On veut écrire un programme permettant de gérer une bibliothèque. Pour chaque livre, on a besoin des informations suivantes:

- titre du livre (sur 150 caractères),
- nom de l'auteur (sur 100 caractères),
- année de parution

Question 1: *Bibliothèque – version 1*

Écrivez le type `livre` permettant de représenter un livre, ainsi que le type `bibliotheque` permettant de stocker au maximum 1000 livres. Dans cette implémentation, on passera par un tableau.

.....
.....
.....
.....
.....

Écrire la fonction `initialiser_bibliotheque` qui permet de créer une bibliothèque vide.

.....
.....
.....
.....

On veut écrire une fonction qui permette la saisie d'un livre sur le fichier standard d'entrée et qui ajoute ce livre à la bibliothèque. On propose les trois prototypes suivants:

```
void ajout_livre_1(bibliotheque b);  
bibliotheque ajout_livre_2(bibliotheque b);  
void ajout_livre_3(bibliotheque* b);
```

- ajout_livre_1: _____
- ajout_livre_2: _____
- ajout_livre_3: _____

-
-
-
-

-
-
-
-
-
-

[illegible]

On souhaite maintenant gérer la bibliothèque avec une liste simplement chaînée. Expliquez les avantages de cette implémentation par rapport à la précédente

En supposant, que le type `livre` défini plus haut ne change pas, définir la nouvelle implémentation du type `bibliotheque` et des structures éventuellement nécessaires.

Ecrire la fonction `int livre_apres(int n);` qui renvoie le nombre de livres dont la date de parution est postérieure à l'année passée en paramètre.

Ecrire une fonction permettant de supprimer un livre de la bibliothèque (cette fonction ne fera aucune lecture).

Question 3: *Fonction atexit*

La fonction standard `atexit` est définie comme:

```
int atexit(void (*function)(void));
```

Cette fonction enregistre le fait que la fonction `function` doit être appelée lorsque le programme se termine (par exemple en appelant la fonction standard `exit`). Les fonctions enregistrées par `atexit` sont appelées dans l'ordre inverse de leur enregistrement. La fonction `atexit` renvoie 0 si l'enregistrement a réussi et -1 dans le cas contraire. On suppose ici que l'on peut enregistrer au maximum `MAX_FUNC` fonctions par `atexit`.

-
- This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

-
-
-
-
-

La fonction standard `memcpy` est définie comme:

Elle copie **n** caractères de la zone mémoire pointée par **src** vers la zone mémoire pointée par **dest**. Pourquoi cette fonction prend des paramètres de type **void ***:

[illegible]