

Questions de cours :

1. Quel est le point commun entre un *module C* et une *bibliothèque C*? quelle est la différence ?
2. Qu'est-ce qu'un *effet de bord* ?
3. Quel est le principal intérêt de l'outil *GNU make* ?
4. Que se passe-t-il lorsqu'un processus reçoit le signal *SIGTERM* ?
5. En combien de parties l'espace mémoire d'un processus est-il divisé ?
6. Pourquoi le type *int* est à proscrire des programmes ?
7. Soit la déclaration suivante : `#define MAXTAB 1024`
Comment s'appelle une telle déclaration ? dans quelle zone mémoire est-elle stockée ?

Exercice : Gestion de la mémoire

Il vous est demandé d'écrire un type représentant un entier au sens mathématique. A la différence des entiers « informatiques » (`char`, `short`, `int`, `long`) les entiers mathématiques n'ont pas de limite de stockage : ils peuvent représenter un nombre entre moins l'infini et plus l'infini. Par conséquent, le nombre d'octets pris en mémoire n'est pas connu à l'avance. Proposez un type optimisé pour minimiser la place prise en mémoire appelé *MathInt*. Écrivez toutes les fonctions usuelles permettant de manipuler un entier. (initialisation, addition, soustraction, affichage, taille de la zone mémoire prise par l'entier).

Exercice : Entrées-sorties complexes

Écrivez un *programme* permettant de trier un ensemble de lignes de texte selon un ordre lexicographique croissant. Votre programme devra soit trier le contenu du fichier dont le nom est passé en paramètre sur la ligne de commande, soit trier le texte présent sur l'entrée standard. Votre programme devra afficher les lignes de texte triées sur sa sortie-standard.

Exercice : Entrées-sorties simples

A partir de vos résultats dans l'exercice sur la gestion mémoire, proposez deux fonctions permettant de sauvegarder puis de relire des nombres de type *MathInt* dans un fichier.

Exercice : Philosophie : pour le plaisir de passer le temps

Quelle est le point commun entre un bisounours et un programme C orienté système ?