



Examen de Programmation en C - IR1 2008-2009

Polycopiés de cours et notes de TD autorisés Durée: 2 heures

Pour chaque fonction que vous écrirez, vous devrez définir soigneusement le prototype en tenant compte de tous les cas d'erreurs possibles, et vous mettrez tous les commentaires nécessaires pour que l'on sache comment il faut l'utiliser.

Exercice 1: bonneteau pâtissier (6 points)

Pour Noël, votre beau-frère Robert veut lancer un business de calendriers de l'avent avec un chocolat différent à découvrir chaque jour. Comme il a lu tout Dolto, il sait que les enfants n'aiment pas avoir le même chocolat que le voisin. Il veut donc pouvoir changer l'ordre des chocolats pour chaque calendrier fabriqué. Avez-vous le cœur de refuser de l'aider ?

1) Écrire une fonction **melanger** prenant un tableau d'entiers (les valeurs représentent les chocolats) ainsi que sa taille. Cette fonction devra mélanger les valeurs contenues dans le tableau en utilisant l'algorithme suivant:

- tant que $taille > 1$ faire
 - tirer au sort un indice X compris entre 0 et $taille-1$
 - échanger la case X avec la case $taille-1$
 - diminuer $taille$ de 1

On supposera que le générateur aléatoire a été initialisé correctement.

2) Adapter la technique précédente pour mélanger un tableau à deux dimensions avec la fonction **melanger2D** qui prend le tableau d'entiers (de type **int****) et ses deux dimensions. Attention: si vous réutilisez la fonction de l'exercice précédent, vous risquez de ne pas avoir un mélange satisfaisant.

3) Écrire une fonction **comparer** qui prend 2 tableaux d'entiers à deux dimensions qui sont supposés être de même taille ainsi que les deux dimensions en question et qui teste si les 2 tableaux sont identiques.

Exercice 2: brouiller l'écoute (6 points)

Grand cinéphile devant l'éternel, votre beau-frère Robert a stocké sur son disque dur les œuvres complètes de Clara M., une jeune actrice dont il ne peut parler sans émotion. Conscient que d'aussi poignants chefs-d'œuvres ne sont pas à la portée intellectuelle de tous, il souhaite disposer d'un

programme qui permette de crypter ces fichiers. Le principe est le suivant: décaler circulairement tous les bits du fichier vers la gauche. Exemple pour un fichier de 4 octets décalé de 3 bits:
01100100 11011110 00010010 01110101

devient:

00100110 11110000 10010011 10101011

1) Écrire une fonction **split** qui prend un octet non signé **x** ainsi que deux autres octets non signés **left** et **right** passés par adresse, ainsi qu'une valeur **N** correspondant au nombre de bits du décalage (**N** est supposé compris entre 1 et 7). Cette fonction devra remplir **left** et **right** avec leurs valeurs correctes correspondants au décalage demandé. **left** devra être décalé correctement vers la droite (note de l'éditeur: c'est exprès que l'une des valeurs est décalée et pas l'autre).

Exemple: pour **x=00100110** et **N=3**, on veut obtenir:
left=00000001 et **right=00000110**

2) En utilisant la fonction précédente (si vous ne l'avez pas écrite, faites comme si), écrire une fonction **crypter** qui prend deux chaînes de caractères **src** et **dest** représentant les noms du fichier à encoder et du fichier résultat, ainsi qu'une valeur **N** correspondant au décalage. La fonction devra réaliser l'opération complète de cryptage, et ce, sans planter en cas de fichier de taille ≤ 1 .

Exercice 3: ça rentre et ça sort (10 points)

À force d'acheter tous les gadgets à la mode, votre beau-frère Robert est aujourd'hui obligé de surveiller ses comptes s'il ne veut pas se faire saisir son 4e écran plasma, celui dans l'escalier qui permet de ne pas perdre le fil du film quand on passe de la cuisine à la chambre. Pour cela, vous devrez fournir quelques outils pour informatiser sa gestion déplorable.

1) Robert a de gros doigts pleins de mayonnaise qui ratent souvent les touches visées sur son clavier, ce qui fait que ses données ne contiennent pas que des nombres. Écrire une fonction **test_reel** qui prend une chaîne de caractères et qui teste si cette chaîne représente bien un nombre flottant. Si c'est le cas, la fonction doit permettre d'obtenir ce réel. Faites bien attention 1) au prototype et 2) à ne pas considérer comme valide une chaîne commençant par un réel mais suivi d'autre chose comme par exemple **-45.454kjhdg**. Indice: si vous utilisez **sscanf** correctement, la solution prend 2 lignes.

2) Pour avoir l'air vaguement sérieux auprès du contrôleur fiscal Lucien Cheval venu lui souhaiter la bonne année, Robert voudrait avoir de belles données bien présentées en mode texte. Pour cela vous utiliserez votre maîtrise de **printf** pour afficher proprement les données contenues dans un tableau de **float** à deux dimensions. Votre fonction **print_tab** prendra le tableau (de type **int****) et ses dimensions, et affichera son contenu sur la sortie standard en respectant les règles suivantes:

- la largeur d'une colonne vaut la largeur de son plus large élément plus un espace à droite et un espace à gauche;
- les autres données de la colonne doivent être alignées à droites;
- les colonnes sont séparées par le caractère |
- les nombres sont affichés avec au moins une décimale, mais pas plus si ce n'est pas

nécessaire.

Par exemple, le tableau suivant:

```
2 7.9 8451
45 13248.677 877
122336.46546 45 44
1 4 9
```

doit être affiché de la façon suivante:

```
|           2.0 |           7.9 | 8451.0 |
|           45.0 | 13248.677 | 877.0 |
| 122336.46546 |           45.0 | 44.0 |
|           1.0 |           4.0 | 9.0 |
```

Astuce: **printf** avec **null** comme premier paramètre permet de connaître le nombre de caractères d'un affichage sans effectuer celui-ci réellement.

3) Afin d'éviter les 4 ans de prison qui l'attendent si ses fraudes sont découvertes, Robert a besoin de se fabriquer de fausses notes de frais de restaurant. Sachant qu'une note de frais est constituée d'une date, d'un montant, d'un nom de restaurant et de plusieurs convives, qu'un convive est caractérisé par ses nom et prénom ainsi que le nom de sa société, proposez une structure de données pour représenter une note de frais (vous pouvez utiliser des sous-structures si nécessaire).

4) Écrire la fonction **creer_note_de_frais** prenant en paramètre une structure représentant une note de frais passée par adresse et toutes les informations nécessaires. Naturellement, le nombre de convives étant variable, la fonction devra être variadique. Elle devra initialiser correctement la note de frais, en signalant une erreur en cas de problèmes dans les descriptions des convives (pas de convive, convive pour lequel il manque des informations). Vous apporterez un soin tout particulier au commentaire accompagnant la fonction pour expliquer comment on l'utilise.