



Devoir surveillé du 11 juin 2008 (consolidation)

CSH : Initiation au C et au shell
Première année



Tous documents interdits. Les exercices sont indépendants. La correction tiendra compte de la qualité de la rédaction et de la présentation. Barème approximatif.

★ **Exercice 1. (6pts)** On souhaite écrire une fonction qui compte le nombre d'occurrence de caractères dans une chaîne donnée.

▷ **Question 1.** (2pts) Écrivez la fonction `int compte_un(char *chaine, char car)` comptant le nombre d'occurrences du caractère passé en deuxième argument dans la chaîne passée en premier argument. Ainsi, `compte_un("bonjour",b)` doit retourner 1 tandis que `compte_un("bonjour",o)` doit retourner 2.

▷ **Question 2.** (4pts) Écrivez la fonction `int *compte_tous(char *chaine)` qui retourne un tableau de 26 entiers (que vous devrez allouer), chacun indiquant le nombre d'occurrences de la lettre correspondante.

★ **Exercice 2. (7pts)** Soit un fichier regroupant les résultats de l'euro 2008 sous la forme suivante :

```
1 Suisse Rep.Tcheque 0 1
2 Portugal Turquie 2 0
3 Rep.Tcheque Portugal 1 3
4 Autriche Croatie 0 1
5 Allemagne Pologne 2 0
6 Roumanie France 0 0
7 Pays-Bas Italie 3 0
8 Espagne Russie 4 1
9 Grèce Suède 0 2
```

Les deux premières colonnes sont les équipes ayant joué le match (dans un ordre non spécifié) et les deux suivantes sont les scores respectifs de ces équipes. Les colonnes sont séparées par une seule espace.

▷ **Question 1.** (½ pt) Écrire une commande shell donnant tous les résultats concernant la France.

▷ **Question 2.** (½ pt) Écrire une commande donnant le score du match entre la France et la Roumanie.

▷ **Question 3.** (2pts) Écrire un script shell listant tous les matches s'étant soldés par un match nul (même nombre de buts inscrits par chaque équipe).

▷ **Question 4.** (4 pts) Écrire un script shell prenant deux noms d'équipes en paramètres et indiquant si la première équipe a gagné, si c'est la seconde ou si le match s'est soldé par un nul.

★ **Exercice 3. (6pts)** Le ROT13 ("rotate by 13 places") (une variante du chiffre de César) est un algorithme très simple de chiffrement de texte. Comme son nom l'indique, il s'agit d'un décalage de 13 caractères de chaque lettre du texte à chiffrer. [...] L'avantage de ROT13, c'est le fait que le décalage soit de 13. L'alphabet comporte 26 lettres, et si on applique deux fois de suite le chiffrement, on obtient comme résultat le texte en clair. Pour cela on doit considérer l'alphabet comme circulaire, c'est-à-dire qu'après la lettre Z on a la lettre A, ce qui permet de grandement simplifier son usage et sa programmation puisque c'est la même procédure qui est utilisée pour le chiffrement et le déchiffrement. (D'après Wikipédia).

Caractère non-chiffré	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Caractère chiffré	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M

▷ **Question 1.** (3pts) Écrivez une fonction `void rot13(char *chaine)` appliquant l'algorithme ROT13 à la chaîne passée en argument (modifiez la chaîne en place pour plus de simplicité).

▷ **Question 2.** (3pts) Écrivez un programme `rot13` prenant deux arguments. Le premier est le nom d'un fichier à crypter par ROT13, et le second est le nom du fichier dans lequel écrire le résultat.

Indication : la fonction `int getline(char **ligne_ptr, int *lgr_ptr, FILE *flux)` permet de lire un fichier ligne à ligne comme le montre l'extrait de code suivant :

```
1 char * ligne = NULL;
2 size_t longueur = 0;
3
4 while (getline(&ligne, &long, in) != -1) {
5     // utiliser ligne
6 }
7
8 // libérer ligne
```

Si `ligne == NULL`, la fonction alloue automatiquement un espace mémoire avec `malloc()` et fait pointer `ligne` dessus. `longueur` indique la taille du bloc. Si `ligne != NULL`, cela doit pointer sur un bloc de taille `longueur`. `getline` peut alors utiliser cet espace, ou allouer un bloc plus gros au besoin (auquel cas, `ligne` et `longueur` sont mises à jour).

Tout cela pour dire qu'il suffit de mettre le code utilisant la `ligne` lue à l'intérieur de la boucle `while`.