

TP 6 : Révisions

Programmation en C (LC4)

Semaine du 5 mars 2007

Ce TP sera « corrigé » par l'ordinateur. L'objectif est de vous donner une note indicative correspondant (très approximativement) à votre niveau actuel. Pour que tout fonctionne correctement, il vous suffit de nommer vos fichiers et les fonctions comme indiqué dans l'énoncé. Vous devez créer un fichier par exercice. Pour connaître votre note, téléchargez le fichier suivant :

<http://www.di.ens.fr/~labatut/LC4/tp6.tar.gz>

Extrayez les fichiers de cette archive dans votre répertoire de travail (celui où vos fichiers `.c` se trouvent) et lancez la commande :

```
$ make -f notes.mk
```

Attention, il faut bien sûr que vous ayez commenté la fonction `main()` de vos sources (vous pouvez sinon utiliser un même fichier pour tous les exercices, si vous souhaitez tester vos fonctions au fur et à mesure).

Exercice 1 — *tableau.c (très facile)* Écrire une fonction `int compte_zero(int n, int *tab)` qui renvoie le nombre de 0 d'un tableau `tab` de taille `n`.

Exercice 2 — *copie.c (facile)* Écrire une fonction `int *copie(int n, int *tab)` qui alloue un nouveau tableau de taille `n`, y copie le contenu de `tab` et renvoie l'adresse de ce tableau.

Exercice 3 — *fusion.c (moyen)* Écrire une fonction `int *fusion(int n1, int *tab1, int n2, int *tab2)` qui alloue la mémoire nécessaire à un tableau de taille `n1 + n2`, y copie dans l'ordre croissant les éléments de `tab1` et `tab2` (on suppose que `tab1` et `tab2` sont déjà triés) et renvoie l'adresse de ce nouveau tableau.

Exercice 4 — *chaîne.c (facile)* Écrire une fonction `char *copie_chaine(char *s)` qui alloue une nouvelle chaîne de caractère semblable à celle passée en paramètre.

Exercice 5 — *bits.c (moyen)* Écrire une fonction `char *chaîne_binaire(char zero, char un, int n)` qui renvoie une chaîne de caractère correspondant à la représentation binaire de `n` (on suppose que `n` est stocké sur 32 bits). On utilisera le caractère `zero` pour représenter 0 et le caractère `un` pour représenter 1.

Exercice 6 — *pointeur.c (moyen)* Écrire une fonction `int **tri(int n, int *tab)` qui renvoie l'adresse d'un tableau de pointeurs de taille `n`, le tableau pointé par `tab` étant lui aussi de taille `n`. Dans la case `i` de ce nouveau tableau, on mettra un pointeur vers le plus petit élément de `tab` qui est plus strictement grand que `tab[i]`. Si un tel élément n'existe pas, alors on placera le pointeur `NULL` dans cette case.