

TD 3 : pointeurs, tableaux

Programmation en C (LC4)

Semaine du 11 février 2008

Exercice 1 — Retour sur le TP2 Écrire une fonction « `void affichage_binaire(unsigned int m)` » qui affiche la valeur de `m` en binaire (on suppose que les `unsigned int` sont codés sur 32 bits). On affichera les bits de poids faible à droite, par exemple : « 459601 en binaire : 00000000000001110000001101010001 »

Exercice 2 — On s’amuse avec les pointeurs... Compléter le tableau en indiquant les valeurs des différentes variables au terme de chaque instruction du programme suivant (on peut aussi indiquer sur quoi pointent les pointeurs) :

programme	a	b	c	p1, *p1	p2, *p2
<code>int a, b, c, *p1, *p2;</code>	x	x	x	x	x
<code>a = 1, b = 2, c = 3;</code>					
<code>p1 = &a, p2 = &c;</code>					
<code>*p1 = (*p2)++;</code>					
<code>p1 = p2;</code>					
<code>p2 = &b;</code>					
<code>*p1 -= *p2;</code>					
<code>++*p2;</code>					
<code>*p1 *= *p2;</code>					
<code>a = ++*p2 * *p1;</code>					
<code>p1 = &a;</code>					
<code>*p2 = *p1 /= *p2;</code>					

Exercice 3 — Échange de variables Soit `adra` et `adrb` les adresses de deux variables `a` et `b` de type `int`. Écrivez une fonction « `void echange(int *adra, int *adrb)` » qui échange les valeurs de `a` et `b`. Écrire l’appel de cette fonction dans la fonction `main`.

Exercice 4 — Échange de tableaux Écrire une fonction qui échange deux tableaux d’entier `t` et `r`, Écrire l’appel de cette fonction dans la fonction `main`.

Exercice 5 — Concaténation 1 Écrivez une fonction « `int* concat_tab(int n1, int t1[], int n2, int t2[])` » qui prend en arguments deux tableaux et leurs tailles respectives, et qui renvoie leur concaténation.

Exercice 6 — Concaténation 2 Écrivez une fonction « `char* concat_string(char* s1, char* s2)` » qui prend en arguments deux chaînes de caractères et qui renvoie leur concaténation. Vous pourrez utiliser le fait qu’une chaîne de caractères se termine par le caractère `‘\0’`.

Exercice 7 — *Sur les structures* On se donne une structure «livre» :

```
struct lvr{
  char titre[20];
  int cote;
  int prix;
};
```

On veut créer une bibliothèque de plusieurs livres, sous la forme d'un tableau de `lvr`. Écrivez une fonction « `struct lvr * init(int n)` » qui renvoie un tableau de `n` livres, qui alloue la mémoire nécessaire et qui initialise les champs à 0 ou à la chaîne de caractère vide. Comment utiliserez vous cette fonction dans la fonction `main` ?

Écrivez une fonction « `void affiche_bib(int n, struct lvr * b)` » qui affiche un tableau de `n` livres.

Exercice 8 Écrivez une fonction « `void echange_lvr(int i, int j, struct lvr * bib)` » qui fait l'échange de deux livres `i` et `j`.