

TD4

Langage C (LC4)

semaine du 27 février

1 Échauffement

Exercice 1 Complétez 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>	×	×	×	×	×
<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>					

2 Passage de paramètres par adresse

2.1 Avec des pointeurs...

Exercice 2 Soient `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`.

Exercice 3 L'instruction `scanf("%d %d", &n, &m);` lit deux valeurs entières écrites en base décimale et les place dans les variables `n` et `m`. À quoi servent les `&` ?

Exercice 4 Écrivez une fonction `int read_int(int *adr)` qui lit sur l'entrée standard (le clavier, par défaut) une valeur entière écrite en décimal et la stocke à l'adresse `adr`. En pratique, la fonction lira (au moyen de `getchar()`) une suite de caractères entre `'0'` et `'9'` et convertira cette suite en un entier. La valeur de retour de la fonction servira à signaler s'il y a eu une erreur de saisie. La fonction renverra 1 si la valeur entrée était bien un entier et 0 sinon.

Exercice 5 Écrivez une fonction : `int set_max(int tab[], int taille, int *adrmax)` qui parcourt le tableau, stocke à l'adresse `adrmax` la valeur la plus grande du tableau, et renvoie le nombre d'occurrences de cette valeur. Écrivez ensuite un exemple d'instruction qui appelle cette fonction.

Exercice 6 Écrire une fonction

```
char *recherche(char *s, char c)
```

qui renvoie un pointeur vers la première occurrence dans la chaîne `s` du caractère `c` passé en argument. Si ce caractère n'apparaît pas dans la chaîne, la fonction devra renvoyer `NULL`.

Exercice 7 À l'aide de la fonction précédente, écrire une fonction

<pre>int compte(char *s, char c)</pre>
--

qui renvoie le nombre d'occurrences de `c` dans `s`.

3 Arithmétique des pointeurs

Exercice 8 Écrivez deux versions d'une fonction `int my_strlen(char* s)` qui renvoie la longueur d'une chaîne de caractères. La première version doit utiliser la notation des tableaux, et la deuxième l'arithmétique des pointeurs.

Exercice 9 Écrivez une fonction `void inverse(char* s, int n)` qui reçoit un pointeur `s` sur une chaîne de caractères de longueur `n`, et qui inverse la chaîne. Utilisez l'arithmétique des pointeurs.