

Travaux dirigés Numéro 1

Le but de ce TD est de vous familiariser avec le langage C, avec la façon dont il permet de structurer les programmes mais aussi avec l'utilisation des divers opérateurs du langage. Pour cela, plusieurs exercices d'algorithmique et programmation élémentaires vous sont proposés dans une première partie et des exercices plus directement liés aux opérateurs du langage vous sont proposés dans une seconde partie.

1 Algorithmique

1.1 Tri des éléments d'un tableau

Le tri utilisé est le tri bulle dont on rappelle ici l'algorithme. Ce tri comporte plusieurs itérations. A l'issue de la première itération, on est certain que le plus grand élément est bien en fin de tableau. Pour cela, on effectue une boucle sur tous les éléments `i` du tableau et si `tab(i+1) < tab(i)` alors, on intervertit `tab(i)` et `tab(i+1)`. On effectue ainsi plusieurs itérations jusqu'à ce qu'aucun déplacement n'ait lieu. Le tableau est alors trié.

1. Définir un tableau d'entiers de taille fixe et l'initialiser dans le programme principal
2. Ecrire tout d'abord une fonction qui réalise l'affichage du tableau
3. Effectuer enfin le tri du tableau à l'intérieur du programme principal et afficher le tableau trié.

2 Opérateurs bits a bits

1. Ecrire une instruction C qui permet de multiplier un entier par 2 sans utiliser l'opérateur de multiplication
2. Soient deux entiers non signés **a** et **b** ; créer un entier non signé **c** dont le contenu est le suivant : les 2 octets de poids faible de **c** sont les 2 octets de poids fort de **a** et les 2 octets de poids fort de **c** sont le complément à 1 des 2 octets de poids faible de **b**
3. Soient deux entiers non signés **a** et **b** qui diffèrent seulement par le bit 10. Ecrire l'instruction C qui permet de le vérifier.
4. Ecrire la fonction `int bitcount(int n)` qui donne le nombre de bits à 1 de l'entier **n**.

3 A cogiter

Donner le résultat des opérations suivantes :

```
2*(( i/5 ) + (4*(j-3)) % (i+j-2))
i <= j
j != 6
c == 99
5*(i+j) > 'c'
(i>0) && (j<5)
(x>y) && (i>0) && (j<5)
```

sachant que :

```
int i = 8 ;
int j = 5 ;
float x = 0.005 ;
float y = -0.01 ;
char c = 'c' ;
```