

## J1MI2013: Algorithmes et Programmes: feuille 2

### Itérer avec la boucle for

#### Travaux pratiques

Se placer dans le répertoire `~/algoprogram/` (pour cela suivre les indications fournies dans la feuille MémoTP).

Dans ce répertoire, télécharger l'archive `tp02.tar.gz` depuis le site de l'UE  
<http://dept-info.labri.fr/ENSEIGNEMENT/algoprogram/src/>

et en extraire le contenu.

Cette commande va créer un répertoire `tp02` dans votre répertoire `algoprogram` et va y déposer des fichiers dans lesquels vous travaillerez pendant la séance de TP.

Pour cela, se placer dans le répertoire `tp02` et lancer l'éditeur de texte `emacs` en chargeant le fichier correspondant au premier exercice de TP (suivre toujours les indications fournies dans la feuille MémoTP).

#### Exercice 1.

1. Dans le fichier `testExercice1.c` fourni, définir une fonction qui calcule  $f(x) = 3x^2 + 2x - 7$ .
2. Compléter ensuite la fonction `main` pour afficher les valeurs de  $f(x)$  pour  $x$  de 1 à 10.

**Exercice 2.** Dans le fichier `testPuissanceDe2.c` fourni, définir la fonction `puissanceDe2` vue en TD et l'appeler dans la fonction `main`.

**Exercice 3.** Étudier le code des programmes contenus dans le répertoire `tp02/boucles`. Les compiler, le cas échéant analyser les messages d'erreurs et les corriger, puis les exécuter.

Pour le programme `tp02/boucles/loop5.c` le compilateur produit un fichier exécutable mais signale un *warning* (avertissement). Si on ne modifie pas le fichier source, à l'exécution on constate que le programme boucle à l'infini.

Il est possible de forcer le compilateur à arrêter une compilation dès qu'il signale un avertissement, ce qui oblige le programmeur à écrire des programmes qui ne génèrent aucun avertissement. Pour cela il faut utiliser l'option de compilation `-Werror` (comme "warning error"). Compiler à nouveau `loop5.c` en ajoutant cette option.

**Exercice 4.** On considère la suite définie par :

$$\begin{cases} u_0 = 1 \\ \forall n > 0, u_n = 3 * u_{n-1}^2 + 2 \end{cases}$$

Écrire une fonction `int suite(int n)` qui calcule et renvoie le terme  $u_n$  de cette suite.

**Exercice 5.** Écrire une fonction `void afficheTriangleInf(int n)` qui affiche le triangle inférieur de la table de multiplication de taille `n`.

Par exemple pour `n = 6`, on doit obtenir l'affichage :

```
1
2  4
3  6  9
4  8 12 16
5 10 15 20 25
6 12 18 24 30 36
```

**Exercice complémentaire 1.** Écrire une fonction `void afficheCarre(int n)` qui affiche un carré de taille `n`. On donne un exemple pour `n = 6` ci-dessous :

```
# # # # # #
#           #
#           #
#           #
#           #
#           #
# # # # # #
```

**Exercice complémentaire 2.** Écrire une fonction `void affichePyramide(int n)` qui affiche une pyramide de taille `n`. On donne un exemple pour `n = 5` ci-dessous :

```
  #
 ##
####
#####
#####
#####
```