Université Paris-Nord — L1
Institut Galilée — Année 2011-2012
Licence 1 - 1er semestre

# Travaux dirigés 3 : variables impératives et structure de contrôle *if*

L'objectif de ce TD est de vous familiariser avec l'écriture de programmes simples en langage C : calcul artihmétique, affichage de résultats et exécution conditionnelle d'instructions. Il aborde les mêmes notions que le TD 1, mais en utilisant le langage C.

## 1 Déclaration et affectation de variables impératives

### 1.1 Trace de programme en C

Soit le programme suivant :

```
1   /* Declaration de fonctionnalites supplementaires */
2   #include <stdlib.h> /* EXIT_SUCCESS */
3   #include <stdio.h> /* printf */
4
5   /* Declaration des constantes et types utilisateurs */
6
7   /* Declaration des fonctions utilisateurs */
8
9   /* Fonction principale */
10  int main()
11  {
12      /* Declaration et initialisation des variables */
13      int x;
14
15      x = 3;
16      x = x + 1;
17      printf("x = %d\n", x);
18
19      /* valeur fonction */
20      return EXIT_SUCCESS;
21  }
22
23  /* Definitions des fonctions utilisateurs */
24
```

1. Que fait ce programme ?
2. Donner la traduction des instructions aux lignes 15 et 16 en langage amil.
3. Donner la trace du programme C. Pour cela vous utiliserez un tableau comportant 1 colonne pour le numéro de ligne + autant de colonnes que de variables utilisées dans le programme + 1 colonne pour l'affichage éventuel du programme.

# 2 Execution conditionnelle d'instructions : *if*

## 2.1 Majeur ou mineur ?

Soit la variable `age`, contenant l'âge d'une personne. Écrire un programme qui affiche si cette personne est majeure ou mineure.

## 2.2 Exercice type : Le minimum de 3 valeurs

Soient 3 variables `a`, `b`, `c`, initialisées à des valeurs quelconques. Écrire un programme qui calcule et affiche à l'écran le minimum des 3 valeurs.

## 2.3 Exercice type : Quel temps fait-il ?

En vous inspirant du codage du genre vu en cours (1 pour MASCULIN, 2 pour FÉMININ), proposez un codage pour indiquer si le temps est COUVERT, ENSOLEILLÉ ou PLUVIEUX. Écrivez un programme principal qui, étant donné le temps affecté à une variable, affiche le temps qu'il fait.

## 2.4 Exercice type : Dans 1 seconde, il sera exactement...

Écrire un programme principal qui, étant donnée une heure représentée sous la forme de 3 variables pour les heures, `h`, les minutes, `m` et les secondes, `s`, affiche l'heure qu'il sera 1 seconde plus tard. Il faudra envisager tous les cas possibles pour le changement d'heure. Deux exemples de sortie sont :

```
L'heure actuelle est : 23h12m12s
Dans une seconde, il sera exactement : 23h12m13s

L'heure actuelle est : 23h59m59s
Dans une seconde, il sera exactement : 00h00m00s
```

Université Paris-Nord — L1
Institut Galilée — Année 2011-2012
Licence 1 - 1er semestre

# Travaux pratiques 3 : affichages et structure de contrôle *if*

Vous allez mettre tous vos programmes écrits dans ce TP dans le répertoire TP3.

1. À partir du début de votre arborescence, créez le répertoire TP3 : `mkdir TP3`
2. Allez dans ce répertoire pour y mettre des fichiers : `cd TP3`

L'étape suivante est à répéter pour chaque nouveau programme (exo1, exo2 etc..) :

3. Créez un nouveau fichier source pour le langage C ou une nouvelle copie d'un programme existant.

   **Création** `gedit exo1.c &` (vous pouvez utiliser `emacs` ou `kwrite` au lieu de `gedit`)

   **Copie** Il est plus rapide de repartir d'une copie de votre programme `bonjour.c` du TP2 pour éviter de retaper tout le squelette. Dans le terminal :
   ```
   cp ../TP2/bonjour.c exo1.c
   gedit exo1.c &
   ```
   Vous pouvez-aussi ouvrir `bonjour.c` et utiliser la fonction *Enregistrer sous...* de votre éditeur mais attention à enregistrer la nouvelle copie dans le bon répertoire.

Vous pouvez utiliser à tout moment la commande `ls` (list directory) pour voir la liste des fichiers d'un répertoire.

Les trois étapes suivantes seront à répéter autant de fois que nécessaire pour la mise au point de chaque programme (apprenez à utiliser les raccourcis clavier).

4. Après avoir fini d'écrire votre programme, enregistrez le.
5. Créez un programme exécutable à partir de votre fichier source :
   `gcc -Wall exo1.c -o exo1.exe`
6. Quand l'étape précédente a réussi (il faut lire attentivement les messages affichés), exécutez le programme pour vérifier qu'il fonctionne : `exo1.exe` (ou `./exo1.exe`).

## 1 Affichage

1. Écrire un programme `exo1.c` qui affiche à l'écran « coucou ».
2. Modifier ce programme pour qu'il affiche à l'écran « coucou » sur cinq lignes de deux façons :
   - avec cinq printf ;
   - avec un seul printf.
3. Écrire un programme `exo2.c` qui affiche à l'écran l'évaluation de l'expression `7 * 3 + 2`.
4. Modifier ce programme pour qu'il affiche à l'écran l'évaluation de l'expression `3 * x + 2`, avec la variable entière `x` initialisée à une valeur quelconque.

## 2 Faut-il répudier la dette ?

Un arbre de décision [1] est un graphe particulier où les nœuds sont des questions et les arêtes sont les réponses à ces questions. Il se lit de haut en bas. On avance dans l'arbre en

1. (cf. `http//fr.wikipedia.org/wiki/Arbre_de_décision`) Les arbres de décision sont très utilisés en informatique pour prendre des décisions automatiquement. Ils sont soit programmés par un humain soit appris automatiquement par un algorithme d'apprentissage automatique.

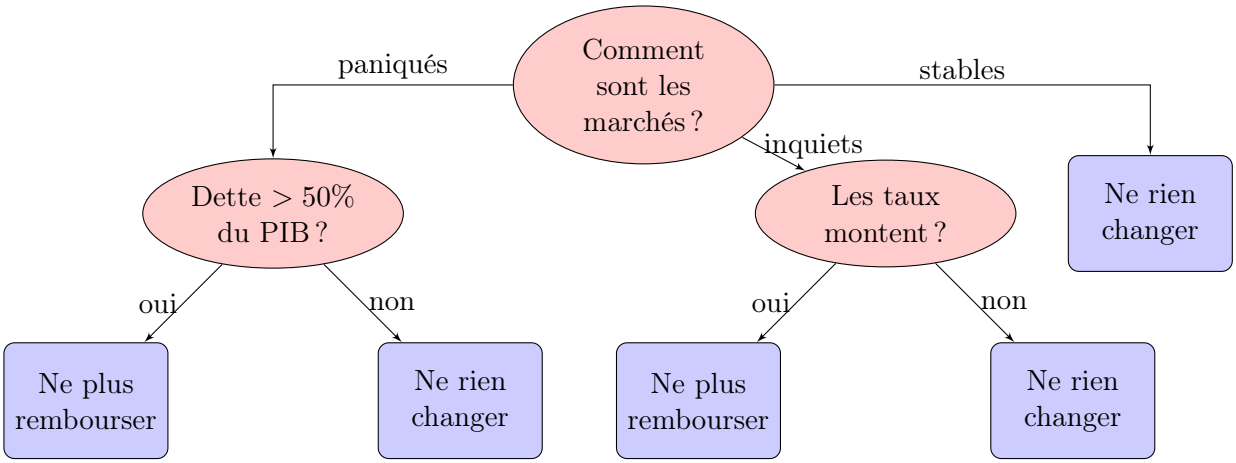


FIGURE 1 – Décider s'il faut continuer de rembourser la dette

répondant aux questions. Les nœuds les plus bas jouent le rôle particulier de classes de réponse au problème initial.

Ici, il y a deux classes de réponse : *« Ne plus rembourser »* et *« Ne rien changer »*. Par exemple, si les marchés sont paniqués et si la dette est strictement supérieure à 50% du PIB alors on ne rembourse plus.

Soient 3 variables entières représentant les propriétés du jour courant pour prendre la décision :

- `marches` est la variable représentant l'état des marchés, elle contient une valeur pour PANIQUE, une pour INQUIETUDE et une pour STABILITE.
- `dette` est la variable représentant la dette en points de PIB.
- `hausse` est la variable représentant le fait que les taux montent ; elle contient une valeur pour NON et une pour OUI.

Écrire un programme `exo3.c` implantant l'arbre de décision pour proposer une réponse étant donné un jour. Après chaque test effectué sur une variable, vous afficherez sa valeur afin de suivre la progression dans l'arbre.

**La méthode à suivre :**

1. Se donner des exemples de problèmes et les résoudre à la main.
2. Écrire un algorithme en français permettant de résoudre ces problèmes.
3. Traduire l'algorithme en langage C, en utilisant l'algorithme pour les commentaires.
4. Tester le programme sur les exemples pour s'assurer de sa correction.