
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.

Correction. Durée 1/2 heure?

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */
```

```

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* affiche coucou sur une ligne 5 fois */
    printf("coucou\ncoucou\ncoucou\ncoucou\ncoucou\n");

    /* valeur fonction */
    return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */

/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int x = 3;

    printf("x * 7 + 2 = %d\n", x * 7 + 2);

    /* valeur fonction */
    return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */

```

2 Faut-il répudier la dette ?

Un arbre de décision¹ 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 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.

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.

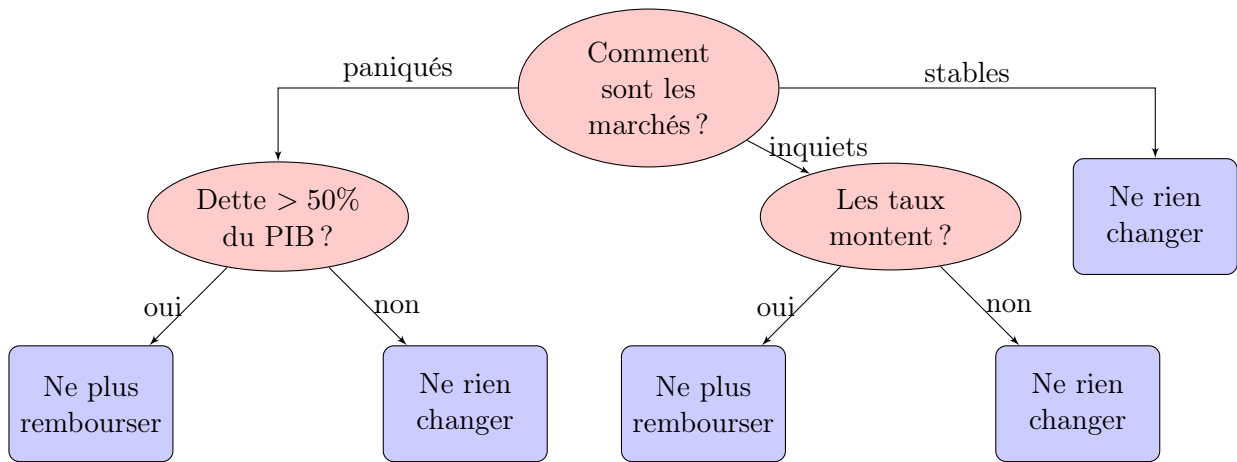


FIGURE 1 – Décider s’il faut continuer de rembourser la dette

- **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.

Correction. Durée 1 heure ?

Les `printf` indiquant les tests effectués sont à ajouter.

```

algo :
si panique alors
    si dette > 50 alors
        ne plus rembourser
    sinon
        ne rien changer
si inquietude alors
    si les taux montent alors
        ne plus rembourser
    sinon
        ne rien changer
si stabilite alors
    ne rien changer
  
```

Les choix sont mutuellement exclusifs.

```

/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */
#define STABILITE 0
#define INQUIETUDE 1
#define PANIQUE 2
  
```

```

#define OUI 1
#define NON 0
/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int marches = PANIQUE;
    int dette = 54; /* pourcents de PIB */
    int hausse = OUI;

    /*
    si marches paniques alors
        si dette > 50 alors
            ne plus rembourser
        sinon
            ne rien changer
    si marches inquiets alors
        si taux en hausse alors
            ne rembourse plus
        sinon
            ne rien changer
    si marches stables alors
        ne rien changer
    */

    /* cas mutuellement exclusif */

    if(marches == PANIQUE)
    {
        if(dette > 50)
        {
            /* ne plus rembourser */
            printf("Ne plus rembourser.\n");
        }
        else /* dette <= 50 */
        {
            /* ne rien changer */
            printf("Ne rien changer.\n");
        }
    }

    if(marches == INQUIETUDE)
    {
        if(hausse == OUI)
        {
            /* ne plus rembourser */
            printf("Ne plus rembourser.\n");
        }
        else /* NON pas de hausse des taux */
        {
            /* ne rien changer */
            printf("Ne rien changer.\n");
        }
    }
}

```

```
    if(marches == STABILITE)
    {
        /* ne rien changer */
        printf("Ne rien changer.\n");
    }

    return EXIT_SUCCESS;
}

/* implantation de fonctions utilisateurs */
```