

## PRATIQUE DE LA PROGRAMMATION TP N°1

### Objectif :

Le but de cet EC comme son nom l'indique et de ce premier TP est de vous amener à découvrir le monde de la programmation par la pratique. Vous êtes sans doute novice dans ce domaine mais pas d'affolement. Lors de cette première séance nos objectifs communs seront les suivants :

- 1- vous aider à appréhender l'environnement de développement que nous utiliserons pour l'instant : Microsoft visual C++ sous Windows
- 2- vous initier aux principes de la création de projets, du chargement, de la compilation de la modification de code source, du test de programmes

### Ce qu'il vous sera demandé :

En dehors du fait d'être attentif et consciencieux dans votre travail, il vous sera demandé de réaliser un compte-rendu à rendre à l'enseignant à la fin de la séance. Ce dernier devra comporter toutes les réponses que vous aurez obtenues aux différentes questions qui vous seront posées dans l'énoncé ci-dessous. Toute question nécessitant de votre part la rédaction d'une réponse dans ce compte-rendu sera identifiée par le symbole : 

### ENONCE :

1) Après avoir ouvert votre session sous Windows, téléchargez les 2 fichiers, tp1.c et tp1\_ajout.c, nécessaires à ce premier TP à l'adresse suivante :

[http://calamar.univ-ag.fr/uag/ufrsen/coursenligne/vpage/new\\_site/](http://calamar.univ-ag.fr/uag/ufrsen/coursenligne/vpage/new_site/)

Faites **attention au dossier** dans lequel le fichier sera rangé (ce doit être un dossier que vous aurez créé dans une de vos zones de stockage personnel en vue d'y ranger vos travaux, non un dossier appartenant au système ou à Microsoft visual C++)

**2) Lancer** l'application Microsoft visual C++.

A titre d'information pour travailler chez vous, vous pouvez utiliser l'environnement Dev-C++ qui au contraire de Microsoft visual C++ (sic !!), est un environnement gratuit de développement en C et C++ pour Windows qu'on peut télécharger librement depuis le site <http://www.bloodshed.net/> (téléchargements : [www.bloodshed.net/dev/devcpp.html](http://www.bloodshed.net/dev/devcpp.html)).

Vous pouvez trouver de nombreux tutoriels sur Internet qui expliquent comment installer et utiliser cet environnement de développement. Nous vous fournirons ici juste ce qui est nécessaire pour votre utilisation courante de Microsoft visual C++ durant ce TP.

PS : les commandes indiquées dans ce qui suit sont valables à la fois sous DEV C++ et Microsoft visual C++. Il est possible qu'elles soient simplement écrites en anglais. Alors, si nécessaire à vos traducteurs !!!

**3) Dans l'environnement, ouvrez le fichier tp1.c, par la commande Ouvrir Projet ou Fichier... du menu Fichier**

**4) Compilez** votre programme à l'aide d'une des commandes du menu **Exécuter** :

**Compiler, Compiler le fichier courant, Compiler & Exécuter** ou **Tout Reconstruire** (dans le cas d'un unique fichier source, toutes ces commandes en produisent la compilation).

Compiler un programme permet de déceler d'éventuels problèmes de syntaxe, erreurs de frappe, etc. A l'issue de cette phase de compilation les erreurs décelées sont indiquées dans une fenêtre en bas de l'écran. En double-cliquant sur un message d'erreur on obtient

l'affichage, dans la fenêtre principale, du texte de l'erreur signalé par une couleur spéciale et une marque dans la marge.

5) Pour **exécuter** le code d'un programme dans Microsoft visual C++ il suffit d'utiliser une des commandes **Exécuter** ou **Compiler & Exécuter** du menu **Exécuter**.

Prenez 5 minutes afin d'observez quelques exécutions du code tp1.c que vous venez de compiler.

👉 Que fait le programme dont le code contenu dans le fichier tp1.c vient d'être compilé ?

6) Le programme que vous venez d'exécuter est codé en langage C.

La règle de structuration d'un programme en C est la suivante : il peut y avoir plusieurs fonctions, dont une est la fonction principale (on dit encore programme principal) et dont le nom est main. Le code d'une fonction est délimité comme suit :

```
type_valeur_de_retour Nom_de_la_fonction(type_valeur_dentrée1 entrée1 , ...)  
{  
    code source de la fonction  
    .....  
}
```

La première ligne où figure le nom de la fonction s'appelle l'en-tête de la fonction. Le code source proprement dit de la fonction, dont les types (ou natures) des valeurs d'entrée et de sortie sont indiqués dans l'en-tête, est compris entre { et }.

👉 Le code source a été découpé en plusieurs fonctions. Chaque fonction étant dédiée à une tâche particulière. Indiquez l'en-tête complet de chacune des fonctions définies dans le fichier tp1.c

👉 Quels sont les 2 règles syntaxiques utilisées ici pour insérer des commentaires dans le code source du programme?

7) Prenez le temps de lire et d'essayer de comprendre le code source de la fonction principale incluse dans le fichier tp1.c que vous venez d'exécuter. Aux vues de votre analyse de néophyte, selon vous :

👉 En C, quelle fonction (quel mot-clé) permet d'afficher un message quelconque à l'écran ?

👉 Relevez et indiquez combien de lignes de code correspondent à des tests de conditions.

8) Vous allez maintenant modifier le code du programme tp1.c ou plus précisément modifier la fonction d'affichage du plateau pour obtenir un résultat « plus joli ». Pour cela, il va vous falloir ouvrir le fichier tp1\_ajout.c, en copier le contenu, puis remplacer le code source de la fonction d'affichage du plateau par celui de la fonction comprise dans le fichier tp1\_ajout.c.

Enregistrez le résultat de votre modification sous le nom tp1\_2.c Pour cela, utilisez la commande **Sauvegarder Sous...** du menu **Fichier**.

Compilez ce nouveau fichier tp1\_2.c puis observez quelques exécutions du programme

👉 En quoi le programme dont le code est contenu dans le fichier tp1\_2. Diffère du précédent ?

9) Voilà le code de la boucle d'affichage :

```
// Une boucle pour afficher chacune des lCourse cases
for (i=1; i<= lCourse ;i++)
{
    // Ici on affiche la ieme case.

    if ( i!= pTortue && i != pLievre )
    {
        // Si il n'y a rien
        printf("_");
    }
    else
    {
        if (i==pTortue) // Si on est a la position de la tortue
        {
            if (i==pLievre) // Tortue et Lievre sont a la meme place
                printf("A");
            else
                printf("T");
        }
        else // Il n'y a pas de tortue (donc il y a un lievre)
        {
            printf("L");
        }
    }
}
```

Essayez de comprendre cet échantillon de code source. Selon vous :

- 👉 A quoi correspondent les symboles encadrés ?
- 👉 Pouvez vous deviner et indiquer la signification de chacun de ces symboles ?
- 👉 A quoi correspondent chacun des 4 termes surlignés en gris ?

10) Ajoutez du code de façon à ce que l'affichage du plateau affiche le nombre de tours de boucles écoulés depuis le début de la course. Enregistrez le code modifié par vos soins dans le fichier tp1\_3.c, compilez le et testez le.

- 👉 Indiquez la nouvelle version de la fonction d'affichage du plateau modifiée par vos soins.