

TP 1 : INTRODUCTION AU C

TABLE DES MATIÈRES

But	1
Installation de votre espace de travail visual C++	1
Utilisation de Visual C++	2
Création d'un fichier source	2
Projet actif	2
Sauvegardes	2
Autres commandes	2
Exercice 1 : Affectation et affichage	3
Exercice 2 : Conversion d'une date en secondes	3
Exercice 3 : Conversion Celcius-Fahrenheit	4

BUT

Vous devez maîtriser à la fin de cette séance les notions suivantes : la déclaration des variables, leur affectation, l'affichage et la saisie de données à l'aide de `printf` et `scanf`, les opérateurs de calcul élémentaires, les conversions.

INSTALLATION DE VOTRE ESPACE DE TRAVAIL VISUAL C++

Avant toute chose, lancez le gestionnaire de dossiers, en cliquant par exemple sur l'icône "poste de travail", puis tapez dans le bandeau : `\\Groseille\Echange\Inge10\LangageC`. Puis double cliquez sur `Install_C`, cette commande installe l'espace de travail que vous utiliserez durant ces TP sur votre compte (dans `Z:\`).

Nous utilisons le logiciel Visual C++ (version 6.0). Il permet de compiler des programmes écrits en C et en C++, puis de les exécuter, de les déboguer... Pour cette année, nous n'utiliserons que le compilateur C, vos fichiers sources porteront donc l'extension `.c`.

Utilisation de Visual C++. Pour démarrer le logiciel Visual C++, utilisez la barre de menu située en bas de l'écran :

Démarrer → *Programmes* → *Microsoft Visual studio 6.0* → *Microsoft Visual C++ 6.0*.

Une fois Visual C++ démarré, utilisez la barre de menu située en haut de la fenêtre :

File → *Open Workspaces*

Allez dans votre espace personnel (Z:) chercher votre espace de travail qui correspond au premier TP :

Z:\TP_C_RAM\TP1\TP1\

Vous venez d'ouvrir un espace de travail composé de 4 projets, Chaque projet correspond à un exercice du TP.

Création d'un fichier source. Cliquez en bas de la fenêtre sur *FileView*. Vous voyez apparaître une (des) arborescence(s) qui illustre la structure du (des) projet(s) de l'espace de travail. Pour l'instant, nous n'utiliserons que le dossier indiqué par *Source*. La partie droite de la fenêtre est destinée à l'éditeur de texte dans lequel vous pourrez créer votre fichier source (fichier terminé par l'extension .c). Pour créer un fichier source dans le projet courant, faire :

File → *New* → *C++ Source File*

et proposer un nom pour votre programme, par exemple `exo1.c`. Ce fichier est automatiquement rangé dans le dossier Z:\TP_C_RAM\TP1\TP1\exo1\.

Projet actif. Si votre espace de travail comporte plusieurs projets, vous constatez que l'un deux (par exemple `exo1`) a son nom inscrit en gras. C'est le projet actif : les commandes de compilation et d'exécution (*Build* → *Build*, *Execute*) concernent ce projet et non les autres. Pour rendre un autre projet actif (par exemple `exo2`) et désactiver le premier, cliquer avec le bouton de droite sur son nom (`exo2`), puis sur *Set as Active Project*.

Sauvegardes. La commande *File* → *save* sauvegarde le fichier "actif" sur lequel vous travaillez (celui qui est au premier plan sur l'écran). Cliquez sur *save all* pour sauvegarder l'ensemble de votre travail (tous vos programmes).

Autres commandes. Explorez la barre de menu pour trouver comment compiler le fichier actif, construire le fichier exécutable (`exo1.exe`), lancer le programme, le débuser, etc. Utilisez l'aide en ligne en double-cliquant sur un nom de fonction (par exemple `printf`) puis en tapant sur la touche F1.

EXERCICE 1 : AFFECTATION ET AFFICHAGE

Pour utiliser la commande d'affichage `printf` vous devez écrire la ligne suivante au début de votre programme :

```
#include <stdio.h>
```

(1) Compléter le code suivant :

```
#include <stdio.h>

int main()
{
    /* Déclaration */
    int x, y;
    char lettre;
    float a, b;
    /*Affectation*/
    x = y = 4;
    lettre = 'p';
    a = 5.6;
    b = a*2;
    /* A compléter , affichage de b */

    x = y+6;
    b = (float) x;
    /* A compléter , affichage de lettre */

    lettre = lettre * 2;
    /* A compléter , affichage de x,y, lettre, a et b ,
       penser à revenir à la ligne */
    return 0;
}
```

(2) Expliquez le résultat de l'opération `lettre*2`.

EXERCICE 2 : CONVERSION D'UNE DATE EN SECONDES

Pour ce nouveau programme, créez un nouveau projet `conversion` ou `exo2`.

Le but de cet exercice est de convertir un temps composé d'un nombre entier de jours, d'heures, de minutes et de secondes, en un temps en seconde.

(1) Écrire le programme comprenant les étapes suivantes :

- Déclarez les variables jour, heure, minute, seconde.
- Déclarez la variable `temps_total` qui recevra le résultat de la conversion.
- Écrivez sur le terminal un message invitant l'utilisateur à rentrer un jour, une heure etc.
- Saisissez les valeurs à l'aide de la fonction `scanf`.
- Calculez le temps total en secondes.

- Affichez le résultat.
- (2) Modifiez le programme pour faire l'opération inverse : vous demandez à l'utilisateur de saisir un temps en secondes et vous écrivez sur le terminal le temps en jours, heures, minutes et secondes. L'affichage à l'écran devra être de la forme :

```
Temps entrée (en seconde) : ...  
Temps = ... jours, ... heures, ... minutes et ... secondes.
```

EXERCICE 3 : CONVERSION CELCIUS-FAHRENHEIT

Certains pays utilisent le degré Fahrenheit comme unité de mesure des températures. Il est donc utile de pouvoir faire la conversion entre cette échelle et l'échelle Celcius utilisée en France. La formule suivante permet de passer d'une échelle à l'autre :

$$T_f = \left(\frac{9}{5}\right) \times T_c + 32,$$

où T_f est une température exprimée en Fahrenheit et T_c la même température exprimée en Celcius.

- (1) Réalisez un programme qui déclare et initialise une température en degré Celcius `tc`, réalise la conversion, et affiche la température en degré Fahrenheit `tf`.

Pour `tc` et `tf`, on utilisera deux variables de type `float`. L'affichage se fera sous la forme :

```
Température entrée : ... celsius.  
Conversion en Fahrenheit : ... F.
```

- (2) Testez votre programme sur un exemple simple.
- (3) Que se passe-t-il si on déclare `tc` en entier et `tf` en float ? si on déclare `tc` et `tf` en entier ? `tc` en float et `tf` en entier ?
- (4) Vérifiez vos réponses en modifiant votre programme et en le testant sur des valeurs simples.

Indication : 28 degrés celcius correspond à 82,4 degrés Fahrenheit.