

Initiation à la programmation impérative. Langage C

L1PC 1^{er} semestre TP 1

1. Cadre de travail

1.1 Contexte de travail

Les machines se trouvent au 2^{ème} étage du « Petit Valrose » (CRIPS 2^{ème} étage). Les machines sont des Pcs sous Windows XP, avec identification. Au cours de ces Tps, vous aurez à utiliser l'environnement de programmation Dev-C++ téléchargeable par exemple sur :

<http://www.01net.com/telecharger/windows/Programmation/langage/fiches/2327.html>

Pour sauvegarder votre travail vous avez besoin d'une clé USB.

1.2 Se connecter

Connectez-vous sur la machine en utilisant votre login et mot de passe ENT.

1.3 Espace de travail

1.3.1 Clé USB

Insérez votre clé USB dans l'ordinateur et travaillez directement dessus. Organisez votre espace de travail en une hiérarchie de répertoires :

```
TPsC
-   TP1
    Exo1
    Exo2
    .....
-   TP2
    Exo1
    Exo2
    .....
.....
-   TP6
    Exo1
    Exo2
    .....
```

1.3.2 Espace disque du Petit Valrose

Vous pouvez aussi enregistrer votre travail sur l'espace disque qui vous est alloué dans les salles du « Petit Valrose ». Pour cela, votre répertoire personnel est monté sur le lecteur « H:\ ». Vous pouvez alors créer l'arborescence décrite en section précédente.

Vous accédez à votre répertoire personnel en ouvrant l'explorateur Windows et en vous plaçant dans le lecteur H:\

2. Déroulement des séances. L'environnement Dev-C++.

Nous allons maintenant utiliser l'environnement de programmation Dev-C++ pour développer des applications C. Nous utiliserons essentiellement trois composants de Dev-C++ : l'**éditeur de texte** pour écrire les programmes (fichiers .c), le **compilateur** pour compiler le programme développé et générer un programme exécutable et le **débogueur** pour la mise au point (si besoin est). Toutes nos applications seront toujours développées dans un fichier unique d'extension .c.

Pour la suite, vous pouvez aussi bien utiliser l'enregistrement sur le lecteur [H:\](#) à la place de la clé USB.

2.1 Déroulement d'une séance

Voici le déroulement standard d'une séance.

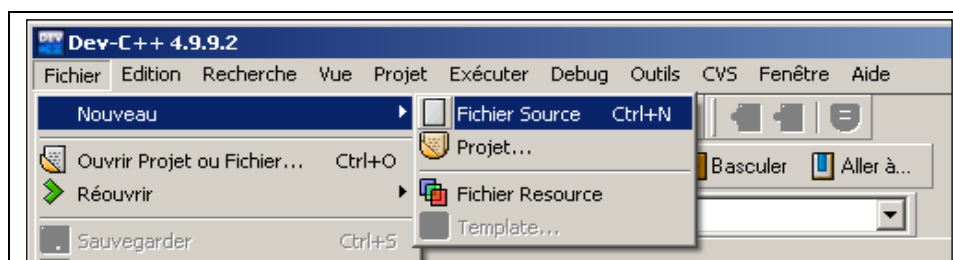
1. Se connecter à la machine
2. Mettre sa clé USB
3.
 - a. Lors de la première séance : créer un répertoire TPsc à la racine de votre clé puis sous TPsc de six répertoires TP1, TP2, ..., TP6. Le répertoire TP1 constitue l'espace de travail pour le premier TP et ainsi de suite.
 - b. A partir de la deuxième séance : utiliser les répertoires TP2, ..., TP7 en fonction du TP du jour.
4. Utiliser Dev-C++ : écriture de programmes, compilation, exécution et sauvegarde du travail dans le répertoire TPi.
5. Retirer la clé USB en sécurité
6. Se déconnecter

2.2 Développement d'une application C avec Dev-C++

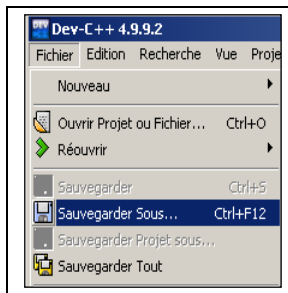
On a deux cas de figure : on développe un nouveau programme **ou** on retravaille avec un programme déjà développé.

Nouveau programme

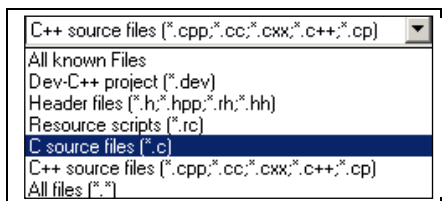
- a. Lancez Dev-C++ : Démarrer-Programmes-Programmations-Dev-C++
- b. Dans le menu : Fichier-Nouveau-Fichier Source



c. Sauvegardez **immédiatement** en faisant Fichier-Sauvegardez sous



Naviguez dans le browser de fichiers pour sauver votre fichier au bon endroit, c'est-à-dire sur votre clé dans le répertoire TPi prévu à cet effet. Donnez-lui un nom « raisonnable ». **Choisissez l'extension .c** puisque nos programmes seront écrits dans le langage C :

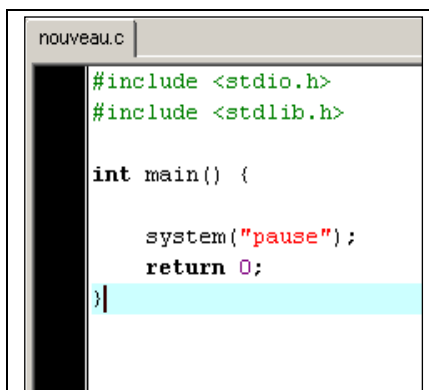


d. Vous disposez d'un onglet d'édition dans laquelle vous allez taper votre code. **Commencez par écrire le squelette de votre programme :**

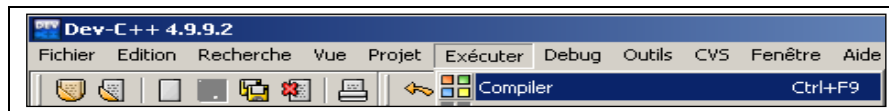
ATTENTION : pour qu'à l'exécution, la console ne disparaisse pas, on écrira avant l'instruction « return 0 ; » du main la ligne :

```
system("pause") ;
```

Ce qui doit donner :

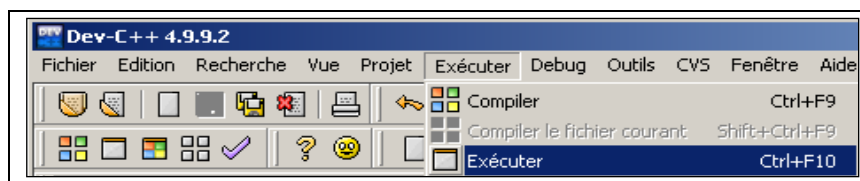


e. **Compilez** ce squelette :



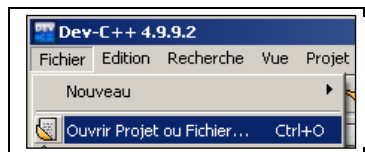
A chaque compilation réussie, un exécutable de même nom que le fichier et d'extension **.exe** est généré dans le même répertoire qui abrite le fichier source **.c**. On peut indépendamment de Dev-C++ lancer cet exécutable en double cliquant dessus.

- f. Pensez à **sauver régulièrement** au fur et à mesure de l'écriture du programme (**Ctrl S**). Après chaque sauvegarde, compilez à nouveau.
- g. Une fois le programme totalement écrit et compilé (sans erreur de syntaxe), **exécutez** votre programme :



Programme déjà développé

Double cliquez sur le fichier **.c** correspondant ou ouvrez ce fichier par :



Puis travaillez comme décrit précédemment.

3. Votre premier programme C

3.1 Ecriture du programme planetes

Utilisez la procédure décrite précédemment pour développer le programme planetes. On nommera le fichier planetes.c et on y recopiera le code donné dans le cours († 1-42).

3.2 Compilation-Exécution

Compilez et **exécutez** ce programme comme indiqué précédemment.

Utilisez le programme pour déterminer quelques surfaces et volumes de planètes du système Solaire.

4. Exercices

Exercice 1 : Rapport de surfaces, de volumes à la Terre

Une fois la surface et le volume de la Terre calculés, modifiez le programme pour qu'il calcule et affiche également le rapport de la surface (respectivement du volume) de la planète à celle (respectivement du volume) de la Terre (pour Mercure vous devriez trouver : rapport des surfaces : 0.146236, rapport des volumes : 0.055922).

Exercice 2 : Masse des planètes

Ajoutez au programme le code nécessaire au calcul et à l'affichage de la masse des planètes en tonnes (on ira chercher sur Internet la densité des planètes du système Solaire). On affichera le résultat en notation exponentielle (%e au lieu de %f) (pour Mercure, densité 5.45, vous devriez trouver environ 3.3×10^{20} tonnes)

Exercice 3 : Vitesse d'un satellite

La vitesse V d'un satellite en orbite circulaire autour de la Terre est donnée par :

$$V = \sqrt{g_0 \times \frac{RT^2}{RT + h}}$$

où g_0 est le champ de pesanteur (9.8 m/s^2), RT le rayon de la Terre (6378 km), h l'altitude du satellite. La racine carrée est donnée par « sqrt » (sqrt(x) est la racine carrée de x). L'utilisation de cette fonction demande l'inclusion du fichier .h qui la définit : math.h.

En vous inspirant du programme planetes, écrivez un programme C (fichier satellite.c) qui calcule la vitesse d'un satellite en fonction de l'altitude. Le programme demandera à l'utilisateur l'altitude (en km), calculera la vitesse à l'aide de la formule donnée puis affichera la vitesse en m/s. Pour 830 km d'altitude (satellite spot) vous devriez afficher 7436.869629 m/s.

Exercices complémentaires

Exercice 4 : Période d'un pendule

La période en seconde des oscillations d'un pendule simple est donnée par la formule $T = 2 \pi \sqrt{l/g}$ où l est la longueur du fil du pendule en mètre et $g \approx 9.8 \text{ m.s}^{-2}$.

Ecrivez un programme qui donne la période d'un pendule à partir de la donnée de la longueur de son fil. Pour une longueur de 1m, vous devriez trouver 2.006073 s.

Exercice 5 : Contraction de la longueur à grande vitesse

Un objet de longueur l_0 au repos (vitesse nulle) a, à vitesse v , la longueur l donnée par la formule $l = l_0 \cdot \sqrt{1 - (v^2/c^2)}$ où c est la vitesse de la lumière (on prendra 300 000 km/s).

Ecrivez un programme qui donne l en fonction de l_0 et v . A partir de quelle vitesse un objet voit sa longueur divisée par 2 (on résoudra le problème en utilisant le programme puis mathématiquement). Pour 1m à 250 000 km/s, la longueur devient 0.553 m.

Exercice 6 : Aire d'un triangle

Si ABC est un triangle, son aire algébrique (c'est-à-dire avec un signe) vaut la moitié du déterminant des vecteurs AB et AC : $0.5 \times (X_{AB} \cdot Y_{AC} - Y_{AB} \cdot X_{AC})$. Ecrivez un programme qui demande à l'utilisateur les coordonnées des points A, B et C, qui calcule l'aire algébrique du triangle et qui l'affiche. Vérifiez qu'avec les points (0 ;0), (1 ;0) et (0 ;1) on trouve 0.5.