

Langage C

Institut d'Optique 1A/S5

TD 0 - Partir sur de bonnes bases

A retenir

Notions étudiées : codage des informations, structure d'un programme simple, bibliothèque standard, spécificités de la fonction main, déclaration et type des variables, affichage à l'écran, formats, commentaires, saisie au clavier.

Bonnes pratiques de programmation : débogage « manuel ».

Diapos à lire :

- LangC-Introduction
- Num-CodageEtOpérations / Num-Calculateur / Num-OSEtProgramme
- LangC-Généralités / LangC-Variables / LangC-ConsoleEcritureLecture
- CodeBlocks-PriseEnMain

Ce premier TD est associée au fichier ***TD0.txt***. Vous devez copier ce fichier d'archives depuis **S:/LangageC/Sources/** dans votre espace de travail (répertoire dans **U:/** - puis dans n'importe quel sous-répertoire).

Exercice 0.

Créez un projet sous Code::Blocks. Pour cela, suivez les instructions de l'enseignant.

Exercice 1.

Que se passe-t-il si vous déplacez le fichier ***main.c*** dans un répertoire nommé ***test*** ? Pouvez-vous déplacer le fichier ***main.c*** dans le répertoire S:/ ?

Exercice 2. Variables et affichage

Vous avez vu le programme de base qui affiche « Hello world ! » à l'écran, lors de l'exercice 0. Vous avez pour cela créé un projet, compilé et exécuté le programme. Dans ce même projet, créez à présent un nouveau fichier source **main2.c** dans lequel vous ferez un copier-coller du programme correspondant à l'exercice 2 du fichier source **TD0.txt**

1. D'après vous, sans l'avoir fait fonctionner, qu'est censé faire ce programme ?
2. Compilez le programme (menu *Build*, « Compile current file ») et analysez le warning signalé dans la fenêtre « Build messages ».
3. Dites alors ce que fait réellement ce programme et ce qu'il affiche.
4. Comment le modifier pour qu'il affiche le bon résultat ? Faites-le !

Pensez à commenter vos modifications...

Exercice 3. Variables et saisie

Créez un nouveau fichier source **main3.c** dans lequel vous ferez un copier-coller du programme correspondant à l'exercice 3 du fichier source **TD1.txt**.

1. Compilez et exécutez le programme.
2. Quel est le rôle de la fonction **scanf** ?
3. Modifiez le programme pour que l'utilisateur sache ce qu'il doit faire et pour afficher le bon résultat !
4. Modifiez ensuite le programme pour que les variables a et b soient des réels.

Exercice 4. Codage

Créez un nouveau fichier source **main4.c** dans lequel vous ferez un copier-coller du programme correspondant à l'exercice 4 du fichier source **TD0.txt**.

1. Compilez le programme (menu *Build*, « Compile current file ») et analysez les différents affichages obtenus.
2. Pourquoi le résultat du dernier calcul (variable d) est faux ?

Exercice 5. Opérations logiques

On se propose à présent de tester les différentes conditions logiques suivantes :

- 86 & 0b00100100
- 86 && 0b00100100
- 86 & 1
- 87 & 1

Créez un nouveau fichier source **main5.c** dans lequel vous ferez un copier-coller du programme correspondant à l'exercice 5 du fichier source **TD0.txt**.

1. Commentez les résultats obtenus dans le premier bloc. Que pouvez-vous conclure sur les opérateurs & et && ?
2. Commentez les résultats obtenus lors des calculs dans le second bloc.
3. Expliquez alors le fonctionnement du dernier bloc.

Exercice 6. Décalages

Enfin, on souhaite tester les différentes opérations suivantes :

- 87 >> 1 87 >> 2 87 << 1

1. Créez un fichier source **main6.c**. Ajoutez les 3 calculs précédents dans ce fichier source et stocker les résultats dans des variables de type **int**. Affichez également les résultats.
2. Compilez et testez votre programme.
3. Comparez aux calculs suivants : $87 / 2$, $87 / 4$ et $87 * 2$. Que pouvez-vous alors dire des calculs précédents ?

Exercice 7. Temps de calcul

On se propose d'utiliser le code fourni pour l'exercice 6 du fichier source *TDO.txt* pour calculer le temps d'exécution d'une instruction.

1. Créez un fichier source *main7.c*. Comparez alors le temps de calcul des opérations précédentes : $87 \gg 2$ et $87 / 4$. Que pouvez-vous conclure ? Le temps d'exécution est-il mesurable en l'état ?
2. Proposez une solution pour mesurer ce temps d'exécution. Le temps obtenu est-il constant ? Expliquez les différences (vous pouvez lancer un autre logiciel en arrière plan pour vérifier).