

Travaux Pratiques 1

Programmation C L2.1

Nous allons voir comment compiler un programme C pour obtenir un exécutable, puis écrire quelques programmes utilisant les formats d’affichage et les instructions conditionnelles.

1. Compilation, création d’un exécutable

Créer un répertoire TP1 et vous y placer. A l’aide de l’éditeur **emacs**, écrire, dans un fichier dont le nom se termine par **.c**, un programme C qui affiche votre nom et votre prénom. Compiler ce programme. Quel est le résultat de la compilation si aucun paramètre autre que le nom du fichier à compiler n’est fourni ? Vérifier que l’exécution est correcte.

Quels sont les droits associés au fichier exécutable créé ? Pourquoi ?

Comment donner le nom **CV** au fichier exécutable.

2. Messages d’erreur, erreurs de conception

Recupérer le fichier **moyenne.c**.

- (a) A l’aide d’un éditeur (**emacs** est conseillé) indenter correctement ce programme;
- (b) compiler avec **gcc** pour obtenir une exécutable de nom **moyenne**
- (c) tester et corriger les erreurs de conception.

3. Multiples de 2 et de 3

Variables, instructions conditionnelles

Ecrire un programme C qui lit deux nombres entiers saisis au clavier, qui affiche ces deux nombres, leur somme, et indique si les nombres sont divisibles par 2 ou par 3.

On désire que sur une ligne, chaque nombre soit écrit sur au moins 3 caractères, suivi de 2 espaces, et, écrit en toutes lettres **divisible par 2** ou **divisible par 3** ou **divisible par 2 et par 3**.

4. Equation de droites dans le plan

- Décrire l’algorithme permettant de déterminer les coefficients a et b de l’équation $y = ax + b$ d’une droite du plan à partir de 2 points n’ayant pas la même abscisse ?
- Ecrire un programme qui affiche l’équation de la droite passant par deux points P et Q dont les coordonnées auront été saisies au clavier.
- Décrire l’algorithme permettant de déterminer le point d’intersection de 2 droites ?
- Transformer le programme précédent pour afficher le point d’intersection de 2 droites.

5. Mise sous forme de siècles, années, mois, jours

Ecrire un programme transformant un nombre de jours J entré par l’utilisateur en nombre de siècles, de mois, de semaines et de jours restant.

Pour simplifier le problème, on considèrera que tous les mois ont 30 jours et toutes les années 360 jours.

L’affichage se fera sous la forme :

J jours correspondent a :

xx siecle xx annee xx mois xx semaine xx jours

On appliquera les règles d'accord du pluriel. ("5 annees" et non pas "5 annee")