

TD - TP 3

Objectif : programmation en langage C

- Boucle, **if**, **else**, **else if**, **switch**, **break** (Chapitre 2)
- Fonction (Chapitre 2, Chapter 4)
- Pointeur (Chapter 4)

1. Plus grand commun diviseur (pgcd)

- (a) Ecrire un programme, contenant seulement une fonction **main**, qui calcule le pgcd de 2 entiers, en utilisant un algorithme itératif basé sur la relation $pgcd(a, b) = pgcd(a - b, b)$, étant donnés deux entiers a et b tels que $a > b$. L'algorithme consiste donc en une boucle dans laquelle le plus grand des deux nombres est remplacé par leur différence, et qui s'arrête lorsque les deux nombres sont égaux, valeur correspondant au résultat recherché.
- (b) Ecrire une autre version de ce programme qui appelle la fonction de prototype :
`int pgcd(int a, int b);`.
Cette fonction renvoie le pgcd de ses deux arguments (selon le modèle de programmation précédent). Mettre les fonctions **main** et **pgcd** dans le même fichier.
- (c) Créer un nouveau fichier exécutable en mettant cette fois les fonctions **main** et **pgcd** dans des fichiers différents. Pour cela, créer un fichier **decl.h** qui contient la déclaration de la fonction **pgcd**.
- (d) Faire une autre version de prototype :
`void pgcdC(int a, int b, int* p_c);`.
Cette fonction renvoie le pgcd de ses deux premiers arguments à l'adresse fournie dans **p_c**. Mettre les fonctions dans des fichiers différents.

2. Résolution de l'équation quadratique $ax^2 + bx + c = 0$

- (a) Ecrire un code qui détermine le nombre **nsol** de solutions réelles différentes et les solutions x_p and x_m

$$x_p, x_m = \frac{-b' \pm \sqrt{b'^2 - ac}}{a}$$

où $b' = b/2$. Dans le cas $b'^2 - ac = 0$, $x_p = x_m$ et **nsol**=1. Si $b'^2 - ac < 0$, il n'y a pas de solution réelle et **nsol**=0. Imprimer **nsol** ainsi que les solutions réelles si celles-ci existent.

- (b) Vérifier la précision des solutions en les substituant dans l'équation quadratique, imprimer l'erreur, c-à-d la différence à 0.
- (c) Ecrire une fonction **rroot** de prototype :
`int rroot(double a, double b, double c, double *p_xp, double *p_xm);`
qui retourne **nsol** et qui calcule x_p, x_m si celles-ci sont définis. Appeler la fonction de **main** qui imprime **nsol** ainsi que les messages
 - i. si **nsol**=0 : "Aucune solution réelle"
 - ii. si **nsol**=1 : "Solution dégénérée xp=xm= ..., erreur=..."
 - iii. si **nsol**=2 : "xp=..., erreur=... xm=..., erreur=..."

Quel est l'ordre de traitement des cas pour rendre le code efficace ?

(d) Bonus :

Traiter le cas $a = 1, b = 2023, c = -1$. Qu'observez-vous ? Quelle en est la cause ? Comment améliorer le code ? Vérifier l'optimisation pour $a = 1, b = -2023, c = 1$.

3. Usage du pointeur de fonction

On revient sur l'exercice d'écriture d'une fonction qui recherche une racine d'une fonction polynomiale par la méthode de dichotomie du TP2, question 3. Ecrire une version plus générale de cette fonction qui prend en argument un pointeur de fonction. Tester cette nouvelle version en utilisant la fonction polynomiale p précédente. L'utiliser ensuite pour calculer une approximation de π à l'aide de la fonction sinus de la bibliothèque mathématique. Faire attention au type de données utilisé. Il faudra déclarer la fonction à l'aide du fichier `math.h` et faire l'édition de liens en ajoutant l'option `-lm` sur la ligne de commande.