

TP1

Préliminaire

Créer le répertoire 'TP-ASD' dans votre répertoire de référence via la commande Unix : « `mkdir TP-ASD` ».

Tous les programmes, documents réalisés lors des TP algorithmique et programmation devront être stocké dans ce répertoire.

Exo1

```
//exo1 du TP1 AS
// auteur : Colette Johnen

#include <iostream>
using namespace std;

int main() {
    int x1, x2, x3;
    cout << ``Saisissez 3 entiers :`` ;
    cin >> x1 >> x2 >> x3;
    if (x1 < 0 || x2 < 0 || x2 > 59 || x3 < 0 || x3 > 59)
        cout << ``erreur de saisie\n`` ;
    else {
        if (x3 == 59) {
            x3=0; x2++ ;
            if (x2 == 60) { x1++; x2++; }
        }
        else x3++;
        cout << x1 << `` `` << x2 << `` `` << x3 << endl;
    }
}
```

Saisissez le programme précédent sous Emacs. Compilez-le, puis testez-le avec les valeurs :

```
2, 16, 80
3, 12, 35
4, 19, 59
5, 59, 0
11, 59, 59
```

Que semble essayer de faire ce programme ?

Renommez les variables, corrigez les bogues.

Recompilez le programme (commande `g++ -Wall tp1.cc -o tp1`)

Modifiez le programme pour placer le bloc « `x3++` » au début du bloc `else` correspondant à « `if (x1 < 0 || x2 < 0 || x2 > 59 || x3 < 0 || x3 > 59)` ».

Ajoutez le(s) code(s) retour(s).

Simplifiez votre programme au maximum.

Exo2

Objectif : écrire un programme qui lit 3 entiers naturels **a**, **b**, **c** et qui affiche les racines du polynôme ax^2+bx+c .

Il faut que la valeur de **a** soit supérieure à zéro pour avoir une équation du second degré.

Si **a** est égale à zéro on a une équation de degré 1 qui a une seule racine : $-c/b$.

Rappel :

Le discriminant du polynôme est le réel $d = b^2 - 4ac$

Si $d > 0$ alors le polynôme a 2 racines réelles

$$r1 = (-b + \text{sqrt}(d))/2a$$

$$r2 = (-b - \text{sqrt}(d))/2a$$

La forme canonique du polynôme est : $a(x-r1)(x-r2)$

par exemple : x^2-4 a deux racines 2 et -2.

La forme canonique du polynôme est $(x+2)(x-2)$

par exemple : $4x^2+5x+1$ a deux racines $-1/4$ et -1 .

La forme canonique du polynôme est $(4x+1)(x+1)$

par exemple : x^2+5x+4 a deux racines -1 et -4 .

La forme canonique du polynôme est $(x+1)(x+4)$

Si $d = 0$ alors le polynôme a une racine double $r = -b/2a$

La forme canonique du polynôme est : $a(x-r)(x-r)$

par exemple : x^2+2x+1 a une racine double -1

La forme canonique du polynôme est $(x+1)(x+1)$

Les étapes :

Ecrivez une fonction/procédure qui saisie 3 entiers sur l'entrée standard.

```
void saisiePolynome2(int& a, int& b, int& c)
```

Ecrivez une procédure qui affiche sur la sortie standard le polynôme

```
void affichagePolynome2(int a, int b, int c)
```

Ecrivez une procédure qui retourne le nombre de racines

```
int calculerNombreRacine(int a, int b, int c)
```

Ecrivez une procédure qui affiche le polynôme dans le format canonique

```
void affichageCanoniquePolynome2(int a, int b, int c)
```

Si vous avez le temps, rajoutez le cas des racines complexes : $d < 0$

Rappel :

Si $d < 0$ alors 2 racines complexes notées

$$r1 = (-b/2a) + \text{sqrt}(-\Delta)/2a i$$

$$r2 = (-b/2a) - \text{sqrt}(-\Delta)/2a i$$

La forme canonique du polynôme est toujours $a(x-r1)(x-r2)$

par exemple : $2x^2+2x+1$ a deux racines : $-0,5 + 0,5i$ et $-0,5 - 0,5i$

La forme canonique du polynôme est $2(x+0,5-0,5i)(x+0,5+0,5i)$

`sqrt` est une fonction de la bibliothèque « `STD` » nommée `cmath`. `sqrt` calcule la racine carrée d'un réel positif.

Nota bene : pour transformer un entier `e1` en un réel en C++ nommé `r1`, utilisez la commande suivante (après avoir déclaré `r1`) :

```
r1 = static_cast<float>e1
```