# Devoir Long

En binôme, écrire un programme en langage C qui résout un des problèmes ci-dessous. Le programme doit être réalisé pendant la séance de TD et le code produit doit être rendu au chargé de TD à la fin de la séance par email ou par clé USB. Les noms des étudiants composant le binôme doivent être inscrits dans le code, sous la forme d’un commentaire.

## Problème 1 – Résolution d’équations de second degré

Etant donné une équation du second degré *ax2 + bx + c = 0*, avec *a*, *b* et *c* fournis par l’utilisateur, trouver les solutions pour cette équation.

Rappel :

Etant donné l’équation *ax2 + bx + c = 0*, le discriminant (delta) est défini comme suit :  
Δ = *b*2 − 4*ac*.

Alors l'équation a deux solutions réelles distinctes *x*1 et *x*2 :

x_1 = \frac{-b + \sqrt{\Delta}}{2a} et x_2 = \frac{-b - \sqrt{\Delta}}{2a}

## Problème 2 – 100% Algèbre

Avec deux entiers positifs a et b, on a effectué les opérations suivantes :

1. L'addition des deux nombres,
2. La soustraction (du plus grand on a retranché le plus petit),
3. La multiplication des deux nombres,
4. La division du plus grand par le plus petit.

La somme de ces quatre résultats a été trouvée égale à 16807.

Ecrire un programme en C capable de trouver les deux nombres initiaux a et b (afficher toutes les solutions possibles).

*Astuce* : il faut tester toutes les nombres dans un intervalle des possibilités.

## Problème 3 – Modulo

Si l'on divise le nombre entier X par: 2, 3, 4, 5, 6, 7, 8, 9,10, 11, 12, le reste est toujours égal à 1. Une autre indication pour vos machines X est inférieur à 30000, et ce n'est pas 1.

Ecrire un programme en C capable de découvrir quel est ce nombre.

*Astuce* : il faut tester toutes les nombres dans un intervalle des possibilités.