

Exercices à faire pendant le TP 2

Exercice 1

Ecrire un programme qui :

- lit au clavier 2 nombres a et b,
- fait une permutation entre les valeurs de a et b,
- affiche leur valeur finale.

Exercice 2

Ecrire un programme qui

- lit au clavier 2 nombres entiers a et b et affiche à l'écran les valeurs qui ont été lues,
- calcule et affiche la somme et le produit de a et b,
- calcule et affiche le résultat de la division euclidienne de a par b (lorsque le calcul est possible), c'est-à-dire leur quotient entier et le reste de la division.

Exercice 3

Ecrire un programme qui affiche les valeurs extrêmes d'une variable de type *int* :

INT_MIN et *INT_MAX*, définies dans *limits.h*.

Exercice 4

Ecrire un programme qui convertit en heures, minutes et secondes une durée exprimée en secondes, donnée par l'utilisateur.

Exercice 5

Calcul de la somme des n premiers entiers :

Ecrire un programme qui calcule la somme des 10 premiers entiers. Utiliserez-vous la même méthode pour la somme des 100 premiers entiers ?

Comment faire si l'on souhaite faire la somme des n premiers entiers, n étant une valeur donnée au clavier par l'utilisateur ? Pour le moment nous ne connaissons pas les itérations en C. La seule façon de faire ce calcul est d'utiliser la formule mathématique qui donne la valeur de cette somme. Ecrivez le programme correspondant.

S'il vous reste du temps en TP :

Exercice 6

Calcul de la "monnaie minimale" :

Ecrire un programme qui, à partir d'un montant entier en euros saisi par l'utilisateur, donne sa décomposition optimale en billets de 500, 200, 100, 50, 20, 10, 5 euros et pièces de 2 euros et 1 euro (ce qui signifie que le montant doit être atteint avec un nombre minimal de billets et de pièces). Pour cela : utilisez une succession de divisions entières afin de déterminer le nombre de billets et de pièces nécessaires pour atteindre le montant.

Ajoutez à votre programme une boucle qui permet à l'utilisateur de faire le calcul plusieurs fois de suite, sans avoir à relancer l'exécution du programme.