

TD 1 : Rappels sur les variables et structures de contrôle

Les algorithmes seront écrits en *pseudo-code* ; on pourra ensuite passer au code C du programme correspondant. On prendra toujours soin d'insérer des commentaires.

Exercice 1

Ecrire un algorithme qui lit le prix HT d'un article, le nombre d'articles et le taux de TVA (5,5% par exemple) et qui affiche le prix total TTC correspondant, puis le prix soldé (-30% par exemple).

Exercice 2

Ecrire un algorithme qui demande deux nombres à l'utilisateur et détermine ensuite si leur produit est négatif ou positif en calculant le produit des deux nombres puis, sans le calculer. Que se passe-t-il quand le produit est nul ? *Facultatif* : modifier l'algorithme pour gérer le cas.

Exercice 3

Nombre secret : écrire un algorithme qui demande à l'utilisateur 1 d'entrer un nombre et à l'utilisateur 2 de le trouver en affichant, à chaque tentative, « trop grand » si le nombre entré est plus grand que le nombre secret, « trop petit » sinon. L'exécution s'arrête quand l'utilisateur 2 a trouvé le nombre secret. *Facultatif* : ajouter une borne sur le nombre d'essais.

Exercice 4

Ecrire le code C du programme qui affiche le texte suivant pour les chiffres de 1 à 10 :

```
1
2    2
3    3    3
...
```

Exercice 5

Ecrire un algorithme qui demande un nombre de départ et puis qui affiche les cinq nombres impairs suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres impairs de 19 à 27. Votre algorithme est-il juste si l'utilisateur entre 18 et non 17 ?

Exercice 6

Ecrire un algorithme qui demande le numéro du mois (entre 1 et 12) à l'utilisateur et affiche le nombre de jours du mois correspondant (on laisse de côté le cas des années bissextiles).

Exercice 7

Ecrire un algorithme qui demande les coefficients d'une équation du second degré et affiche les racines réelles si elles existent. Pour le passage au programme en C, on utilisera la fonction *sqrt* de la bibliothèque *math.h*

Exercice 8

Soit une fonction f sur $[a, b]$ et telle que $f(a) f(b) < 0$.

La méthode de recherche par dichotomie permet de trouver la racine de la fonction f , i.e. la valeur r telle que $f(r) = 0$.

Pour cela, on répète les étapes suivantes, jusqu'à convergence :

1. déterminer $m = (a + b) / 2$ et calculer $v = f(m)$;
2. si $f(m) f(a) > 0$, remplacer a par m , sinon remplacer b par m .

L'algorithme est déjà explicite dans l'énoncé => le coder en C.