

Exercices sur les boucles

Ticket de caisse

L'utilisateur saisit des couples de nombres quantité puis prix unitaire, et termine par un prix ou quantité égal à 0. Calculer et afficher le prix total.

Intersection d'intervalles

Deux intervalles de nombres entiers sont donnés par leurs extrémités : $[a1 ; a2]$ et $[b1 ; b2]$.

Écrire un algorithme qui demande à l'utilisateur de saisir les intervalles, puis affiche si oui ou non leur intersection est vide.

Vérification des parenthèses

Écrire programme qui teste si, dans une chaîne de caractères, les couples parenthèse ouvrante – parenthèse fermante correspondent

- dans une première version de l'algorithme, vous supposerez qu'il n'y a qu'un type de parenthèses, i.e. '(' et ')'
- dans une deuxième version, on va autoriser deux types de parenthèses : '(' ')', '[' ']'
- à faire quand on aura vu les tableaux : dans une troisième version, l'utilisateur pourra saisir les couples de parenthèse ouvrante et parenthèse fermante, autant de couples qu'il veut

Voici des exemples de chaînes avec des parenthèses correctes et incorrectes

"(a(b))c(d)" correct; "a[b[c(d)e] (f[g]h)ij]" correct ; "a[bc(d)e]" incorrect ; "a(b(c)d)" incorrect

Nombre parfait

Écrire un programme qui teste si un nombre donné par l'utilisateur est un nombre parfait. Un nombre n est dit parfait s'il est égal à la somme de ses diviseurs propres. Par ex., 6 est un nombre parfait car $6 = 3*2*1$ et $6 = 3+2+1$. (Les diviseurs propres de n sont ses diviseurs différents de n).

Défi : calcul de la racine carrée

Documentez vous sur un algorithme de calcul de la racine carrée d'un nombre (par ex. celui qui apparaît en premier sur la page de Racine carrée de Wikipedia). Écrire un programme qui calcule la racine carrée suivant cet algorithme. Vous pouvez utiliser la fonction `sqrt` pour tester l'exactitude du résultat.