

TP n°2
structures de contrôle

Variables, expressions

Exercice 1 : Écrire un programme qui demande à l'utilisateur de saisir les notes des matières M1, M2 et M3 puis calcule et affiche la moyenne de ces notes sachant que la matière M1 a un coefficient de 3, M2 a un coefficient de 2,5 et M3 un coefficient de 6.

Exercice 2 : Un pouce (inch) correspond à 2,54 cm. Un pied (foot) correspond à 12 pouces. Un yard correspond à 3 pieds. Un mile correspond à 1760 yards (5280 pieds). Demandez à l'utilisateur de saisir une distance en pouces (un nombre entier de pouces) et convertissez la en unités impériales (nombres de miles, de yards, de feet et d'inches). Votre affichage doit gérer correctement le pluriel des mots. Convertissez aussi cette distance en mètres.

*

Exercice 3 : Écrivez un programme qui demande à l'utilisateur une somme en francs et la convertit en euros ainsi que dans les diverses monnaies de la zone euro (1€=2,0371 FL (Pays-Bas) =0,787564 £IR (Irlande) =6,55957 F (France) =200,482 ESC (Portugal) =166,386 PTA (Espagne) =1936,27 LIT (Italie) =340,750 DR (Grèce) =13,7603 SCH (Autriche) =40,3399 FLUX (Luxembourg) =1,95583 DM (Allemagne) =40,3399 FB (Belgique) =5,94573 MF (Finlande)). L'affichage devra comporter au plus 2 chiffres après la virgule et respecter les règles d'arrondi.

Exercice 4 : Écrire un programme pour aider une caissière à rendre la monnaie. Votre programme doit demander la somme que le client doit ainsi que la somme versée par le client. Il doit alors afficher le nombre de billets de 500€, 200€, 100€, 50€, 20€, 10€ et 5€ ainsi que le nombre de pièces de 2€, 1€, 0,50€, 0,20€, 0,10€, 0,05€, 0,02€ et 0,01€ qu'il faut rendre au client. On supposera que la caisse contient tous les billets et pièces nécessaires et on minimisera bien sûr le nombre de billets ou pièces rendus au client.

Structures conditionnelles

Exercice 5 : Écrivez un programme qui permet de déterminer toutes les solutions d'une équation du second degré à une inconnue ($ax^2 + bx + c = 0$) dont les coefficients a , b et c sont donnés par l'utilisateur.

Exercice 6 : Écrire un programme qui affiche vrai ssi le nombre que tape l'utilisateur est le carré d'un entier.

Pensez-vous que votre programme fonctionnera pour tous les nombres ? Comment faire en sorte de régler les problèmes liés aux erreurs de calcul ?

Modifiez votre programme pour afficher à l'utilisateur une phrase en français qui indique que le nombre est un carré parfait.

Exercice 7 : Écrivez un programme qui écrit en toutes lettres une somme demandée à l'utilisateur et exprimée en euros. On se limitera à des sommes ne dépassant pas 9999,99€. On précise que mille est invariable, que cent et vingt prennent un s quand ils sont **multipliés** et non suivis par un nombre, que l'on met un trait d'union dans l'expression des nombres inférieurs à cent et que dans certains cas, le trait d'union est remplacé par la conjonction ET. Exemples : 108 = cent huit, 97 = quatre-vingt-dix-sept, 180 = cent quatre-vingts, 500 = cinq cents, 583 = cinq cent quatre-vingt-trois, 120 = cent vingt, 21 = vingt et un.

Boucles

Exercice 8 : Écrivez un programme qui demande à l'utilisateur de saisir le numéro d'un mois (de 1 à 12), et l'année. Vous devez ensuite afficher combien de jours contient ce mois. On rappelle que les années sont bissextiles si elles sont divisibles par 4 et pas par 100, ou bien si elle sont divisibles par 400.

Exercice 9 : Dans un jeu vidéo, si un vaisseau spatial dépasse la limite droite de l'écran, il réapparaît du côté gauche. Idem pour les autres directions. Écrivez un programme qui, en boucle, demande à l'utilisateur de saisir les coordonnées

(x, y) du déplacement du vaisseau, puis met à jour ses coordonnées. Vous utiliserez les variables W et H pour désigner la largeur et la hauteur de l'écran (par exemple 1920×1024).

Exercice 10 : Un entrepôt stocke certains produits périssables au bout de quelques mois (disons 4 mois). Pour gérer correctement ces produits, leur code barre se termine par l'année (2 chiffres) et le mois (2 chiffres) de leur fabrication. Écrivez un programme qui demande à l'utilisateur de saisir la date courante, puis des codes barres (en pratique, une douchette sera utilisée) et qui affiche à la fin le nombre de produits qui expirent dans 1, 2, 3 ou 4 mois ou qui sont déjà expirés. Vous devrez aussi générer un graphique ASCII en affichant 1 étoile par produit sur les lignes qui correspondent à un âge de 1, 2, 3, 4 mois, et plus de 4 mois.

Exercice 11 : Écrire un programme qui calcule le PGCD (Plus Grand Commun Diviseur) de deux nombres a et b . Une version particulièrement simple de l'algorithme d'Euclide qui calcule ce PGCD est de soustraire le plus petit nombre (entre a et b) du plus grand et de répéter l'opération aussi longtemps que $a \neq b$.

Exercice 12 : Écrire un programme qui calcule le PGCD (Plus Grand Commun Diviseur) de deux nombres a et b en utilisant une autre version de l'algorithme d'Euclide. Il faut s'assurer d'abord que $a > b$. Ensuite, on effectue une première division entière de a par b : $a = q_0 \cdot b + r_0$. On divise alors le dernier diviseur par le dernier reste obtenu, pour obtenir un nouveau quotient et un nouveau reste : $b = q_1 \cdot r_0 + r_1$. On répète cette opération jusqu'à obtenir un reste nul $r_0 = q_2 \cdot r_1 + r_2, r_1 = q_3 \cdot r_2 + r_3, \dots, r_{n-2} = q_{n-1} \cdot r_{n-1} + 0$. Le dernier reste non nul (r_{n-1}) donne le PGCD.

Ce programme est-il plus rapide ou plus lent que celui de l'exercice précédent ?

Exercice 13 : Écrire un programme qui simplifie une fraction.

Exercice 14 : Écrire un programme qui additionne deux fractions.

Exercice 15 : Écrire un programme pour aider une caissière à rendre la monnaie. Votre programme doit demander (en boucle) la somme que le client doit ainsi que la somme versée par le client. Il doit alors afficher le nombre de billets de 500€, 200€, 100€, 50€, 20€, 10€ et 5€ ainsi que le nombre de pièces de 2€, 1€, 0,50€, 0,20€, 0,10€, 0,05€, 0,02€ et 0,01€ qu'il faut rendre au client. Votre programme doit gérer le nombre de pièces et de billets disponibles dans la caisse.

Exercices en lien avec R1.03 (introduction à l'architecture des ordinateurs)

Exercice 16 : Écrire un programme qui demande à l'utilisateur de saisir un nombre entier positif et qui affiche alors le nombre de chiffres nécessaires pour représenter ce nombre en base 10. Par exemple, il doit afficher 1 pour le nombre 7 et 3 pour le nombre 732.

Exercice 17 : Écrire un programme qui demande à l'utilisateur de saisir un nombre entier positif et qui affiche ce nombre sur une largeur d'exactly L caractères (en ajoutant des espaces pour arriver à cette largeur spécifiée). Vous devez afficher 3 versions du nombre sur L caractères : avec alignement à gauche, avec alignement à droite et avec alignement centré.

Exercice 18 : Écrire un programme qui demande à l'utilisateur de saisir les chiffres d'un nombre entier positif (en base b) puis affiche ce nombre en base 10. b est choisi par l'utilisateur, et pour simplifier, on supposera que $b \leq 10$ et que l'utilisateur saisit les chiffres séparément (par exemple 1 2 3 4 pour le nombre 1234).

Exercice 19 : Écrire un programme qui demande à l'utilisateur de saisir un nombre entier positif (en base 10) puis affiche ce nombre en base b . b est choisi par l'utilisateur, et pour simplifier, on supposera que $b \leq 10$. Vous ne devez pas utiliser de string, de liste ou de récursivité dans cet exercice. Pour remettre les chiffres à l'endroit, vous utiliserez un entier r et quand un nouveau chiffre est calculé, "vous l'ajouterez" à la gauche de l'entier r .

Pouvez-vous trouver une autre méthode pour produire les chiffres dans le bon ordre ?

*