

Remarques et recommandations préalables :

- lorsqu'il est demandé d'écrire du code ou des fonctions, le langage est Java ;
 - le correcteur n'attachera pas une importance excessive à l'absence de quelques points-virgules ;
 - par contre, il n'en sera pas de même, par exemple, en ce qui concerne la rigueur apportée à la déclaration des variables, la syntaxe des structures de contrôle et l'utilisation des accolades { et } ;
 - la présentation (indentations) et les commentaires facilitant la compréhension de la stratégie mise en œuvre par la solution seront également appréciés ;
 - les différents exercices sont indépendants.
-

Exercice 1.

Répondre sur la feuille jointe, où vous écrierez une indication de votre choix que vous reporterez sur votre copie. Il peut s'agir de vos nom/prénom par exemple, à moins que vous ne souhaitiez une correction anonyme de votre copie.

Pour le nombre de 12 chiffres binaires figurant sur cette feuille, il vous est demandé de donner successivement son expression dans les bases dix, seize et huit, puis l'expression en machine de son opposé exprimée sur 16 bits.

Exercice 2.

Écrire un programme qui :

- demande à l'utilisateur d'entrer deux nombres entiers a et b ,
 - vérifie que ces deux entiers sont positifs,
 - calcule et affiche l'aire et le périmètre du triangle rectangle dont les petits côtés ont pour longueur a et b . On pourra pour cela utiliser la fonction `Math.sqrt` qui, appliquée à un paramètre de type `double`, retourne la valeur de sa racine carrée (également de type `double`).
-

Exercice 3.

On s'intéresse à l'opérateur booléen *maj* ternaire (c'est-à-dire à 3 arguments) dont la valeur $maj(a, b, c)$ est *vraie* si et seulement si au moins deux de ses arguments ont la valeur vraie (cette règle est dite de la majorité).

Question 3.1. Donner la table de vérité de cet opérateur.

Question 3.2. Donner une **expression booléenne** Java (c'est-à-dire de type `boolean`) et n'utilisant que les opérateurs logiques de Java, construite avec trois variables `a`, `b`, `c` de type `boolean` et qui a pour valeur de vérité la valeur de $maj(a, b, c)$.

Question 3.3. Écrire une suite d'instructions calculant $maj(a, b, c)$ en utilisant cette fois des `if ... else` imbriqués, sans utiliser d'opérateurs logiques. Le résultat sera stocké dans une variable `maj` de type `boolean`, déclarée en début de code.

Exercice 4.

On suppose qu'un tableau `tab` de `n` entiers contient les `n` premiers nombres entiers (c'est-à-dire `0, 1, ..., n-1`): chaque entier apparaît exactement une fois dans le tableau, mais les nombres n'apparaissent pas nécessairement dans l'ordre.

Écrire un fragment de code construisant à partir de `tab`, un tableau `position` de même longueur que `tab` et initialisant son contenu comme suit :

`position[i]` est égal à la position de l'entier `i` dans `tab`.

Par exemple, pour la valeur `tab={2, 4, 1, 0, 5, 3}`, le tableau `position` contiendra `{3, 2, 0, 5, 1, 4}` après l'exécution du code.

Exercice 5.

On considère n points A_0, A_1, \dots, A_{n-1} de l'espace dont les coordonnées (abscisse x_i , ordonnée y_i et cote z_i) sont regroupées dans trois tableaux

`double[] abscisses, ordonnees, cotes;`

Ces points sont par ailleurs pondérés, c'est-à-dire que chacun possède une masse m_i .

Les masses des différents points sont regroupées dans un tableau

`double[] masses;`

On appelle barycentre de cet ensemble de n points, le point G défini uniquement si

la somme des n masses est non nulle (donc si $m = \sum_{i=0}^{n-1} m_i \neq 0$)

et dont les coordonnées sont alors :

$$x_G = \frac{1}{m} \times \sum_{i=0}^{n-1} (m_i \times x_i) \quad y_G = \frac{1}{m} \times \sum_{i=0}^{n-1} (m_i \times y_i) \quad z_G = \frac{1}{m} \times \sum_{i=0}^{n-1} (m_i \times z_i)$$

Écrire une fonction `barycentre`, qui recevant en paramètres quatre tableaux `abscisses`, `ordonnees`, `cotes` et `masses`,

- affiche les coordonnées du barycentre de l'ensemble donné de points pondérés, s'il existe ;
 - affiche un message d'erreur sinon.
-

Exercice 6.

On rappelle qu'un polynôme $P(x)$ de degré n à une variable x s'écrit :

$$P(x) = \sum_{i=0}^n a_i x^i = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$$

1. On représente un polynôme par un tableau `pol` de nombres réels, le coefficient a_i du terme de degré `i` correspondant à `pol[i]`.

Question 6.1. Écrire une fonction qui, étant donné un tableau `pol` de réels et un réel α , retourne la valeur du polynôme correspondant à `pol` pour la valeur α de `x`.

On pourra utiliser la fonction `Math.pow` qui, appliquée à deux valeurs x et n de type `double`, renvoie la valeur x^n de type `double`.

2. On considère maintenant un polynôme qui satisfait les conditions suivantes :

- il a une valeur strictement positive en 0 et une valeur strictement négative en 1;
- il est strictement décroissant entre 0 et 1.

Question 6.2. Écrire une séquence d'instructions Java qui permettra d'afficher le zéro du polynôme compris entre 0 et 1 (les propriétés du polynôme garantissent son existence) à 0,001 près (c'est-à-dire si $0 < x < 1$ et $P(x) = 0$, on affichera une valeur de x à 0,001 près).

Feuille de réponse à l'exercice 1

Utilisez l'espace disponible pour chaque question
pour justifier, si vous le souhaitez, votre réponse à la question

Code distinctif choisi par l'étudiant :

Code pour la correction : **69**

Nombre binaire de 12 chiffres à traiter : **010100111101**

Question 1.1. Valeur correspondante en base dix :

Question 1.2. Valeur correspondante en base seize :

Question 1.3. Valeur correspondante en base huit :

Question 1.4. Expression de l'opposé du nombre sur 16 bits :