KATA 1 : Basic algorithm - Sums (30 minutes - 5 points)

1. Soit une classe *Maths* qui contient une méthode de classe *squareRoot(int n)* déjà écrite :
    a. Quel est le mot clé pour dire qu'une méthode est une méthode de classe ?
    b. Quelle est la syntaxe pour faire appel à *squareRoot*?
2. Écrire une méthode de classe *ofOdds* dans une classe appelée *Sum* qui va faire la somme de tous les nombres impairs dans une liste d'entiers.
   Signature de la fonction : *public static int ofOdds(List<Integer> list);*
   Exemple :
   Sum.ofOdds(Arrays.asList(1,2,3,4,5)) retournera *9*.
   Sum.ofOdds(Arrays.asList(2,4)) retournera *0*.

KATA 1 : Basic algorithm - Product (30 minutes - 5 points)

1. Dans une classe il peut y avoir des attributs et des méthodes. En mettant le mot *static* devant un attribut ou une méthode, elle devient un attribut ou une méthode dite "de classe", sinon, c'est une méthode d'instance.
   Soit une classe nommé ***"MyClass"*** sans attributs, avec une méthode ***"myMethod"*** sans paramètres :
    a. Si l'on en fait une méthode de classe (*static*), quelle est la syntaxe pour appeler cette méthode ?
    b. Même question, si l'on veut appeler cette méthode sans le mot "static".

2. Écrire une classe *Product*, dans laquelle il y aura la méthode de classe *"ofList"* qui va servir à calculer le produit des nombres dans une *List* d'entiers.
   Signature de la méthode : *public static int ofList(List<Integer> list);*

   Exemple :
   Product.ofList(Arrays.asList(1,2,3,4)) retournera 24
   Product.ofList(Arrays.asList(0,1000, 200, 5)) retournera 0.

KATA 2 HEADS : Level 1 algorithm - Count (30 minutes - 5 points)

Un nouveau séminaire chez HEI va prendre place, on demande dans HEI Tribu la liste des STD des élèves intéressés. Pour arranger les salles, vous avez la tâche de compter le nombre des L1(2022) et des L2 (2021) dans cette liste.

Créez une méthode pour le faire. On prendra en paramètre l'année d'entrée de la promotion recherchée, et la liste de STD des élèves intéressés.

Signature de la méthode :
public static int countByPromotion(int promotionYear, List<String> stdList);

Exemples :
List<String> list = List.of("STD21001", "STD21002", "STD23005", "STD21015", "STD22088", "STD22103");

YourClass.countByPromotion(2021, list) => 3
YourClass.countByPromotion(2022, list) => 2

Rappels : String.valueOf(...) convertit une valeur en String, et Integer.valueOf(...) convertit une valeur en int.

KATA 2 TAILS : Level 1 algorithm - Count (35 minutes - 5 points)

Il y a une certaine tendance qui dit que les élèves de HEI vivent à Analamahitsy pour la plupart. Vérifions si c'est vrai, en comptant par quartier ! On vous fournit donc les adresses de tous les élèves dans une liste de String, et le nom de quartier qu'on souhaite examiner.
Retournez donc combien d'adresses parmi la liste ont le quartier en question dans leur description, et ce, de manière *insensible* à la casse.

Signature : *public static int countByCity(List<String> addressList, String city);*

Exemples :
List<String> cities = List.of("analamahitsy ambany, lot XYZ", "Analamahitsy Cité, Lot IINX", "aNaLaMaHiTsY Tanana, Lot IINZ", "Itaosy, lot ABC", "itaosy-unis, lot GHL");

YourClass.countByCity(cities, "analamahitsy") => 3
YourClass.countByCity(cities, "itaosy") => 2