

Programmation multi-paradigme INF2162

Lundi 24 Janvier 2022 – 2 h

Lisez attentivement tout l'énoncé avant de commencer.

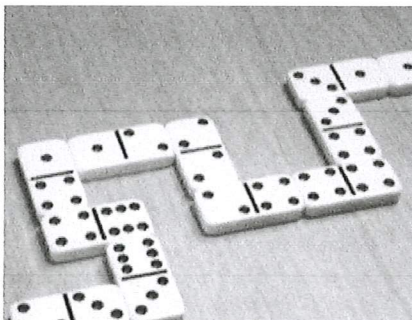
Aucun document n'est autorisé (ou calculatrice / téléphone portable / montre connectée). Le correcteur prendra en compte la pertinence de l'argumentation, des exemples et la qualité de la rédaction (propreté, expression et **orthographe**). En aucun cas une réponse ne peut se réduire à du code non expliqué. Le barème est donné à titre indicatif. Le code sera exprimé dans les langages Java ou Scala au choix si non spécifié par la question.

Question 1 (5 points) : Soit une fonction **F** donnée. On veut calculer le résultat de **F(2) + F(3) + F(5.3) + F(6.2)** de la manière la plus rapide possible en utilisant le parallélisme. Donnez le programme Scala (ou Java comme vous voulez).

Question 2 (8 points) : On a un jeu de dominos qui contient les pièces suivantes : (8,1), (4,8), (3,2), (8,3), (2,5), (3,4), (1,3) : à gauche le nombre de points de gauche, à droite, le nombre de points de droite.

2.1 Comment représentez-vous le type de données d'un domino en Scala ?

2.2 Fabriquez une fonction générique (fonctionnelle) qui permet de créer une liste en en écartant un seul élément :



withoutFirst(List(1,2,6,3,2), 2) donne List(1,6,3,2) (le premier 2 a été écarté). Cette fonction est l'identité si l'élément à enlever n'est pas présent dans la liste.

2.3 Etant donné un domino que vous avez tiré (par exemple (8,1)), vous cherchez maintenant à ordonner les autres dominos pour en faire une série. Par exemple, la série (8,1), (1,3), (3,4), (4,8), (8,3), (3,2), (2,5) : L'idée c'est que le nombre de points de droite d'un domino soit suivi par un domino à droite qui ait le même nombre de points à gauche (et que tous les dominos soient placés). Par exemple, (1,3) peut être suivi par (3,4) ou par (3,2) etc.. Il y a donc plusieurs solutions.

Proposez une fonction récursive (donc qui réalise une recherche – n'utilisez pas un générateur combinatoire comme *Permutation*) qui renvoie la séquence de placement de tous les dominos à partir d'un premier domino que vous spécifierez. Donc par exemple (8,1), (1,3), (3,4), (4,8), (8,3), (3,2), (2,5) si vous commencez par (8,1).

Si ce n'est pas possible, la fonction renvoie une liste vide. Vérifiez que votre fonction trouve la bonne séquence en partant de (4,8), (3,2), (8,3), (2,5), (3,4), (1,3) et avec (8,1) (déroulez sur votre copie la séquence de recherche de votre fonction – ce qui prouvera que ça marche).

Question 3 (4 points) : Expliquez ce qu'est le polymorphisme. Donnez des exemples.

Question 4 (3 points) : Expliquez ce qu'est le concept de surcharge, donnez des exemples.