

# Séance d'exercice 1

CPUMons

8 décembre 2019

**Avant de commencer...** Ce document comporte plusieurs exercices dont la difficulté est renseignée par un nombre et est *globalement* croissante. Il est évidemment fortement recommandé de réfléchir au préalable sur papier avant de se lancer dans la partie implémentation.

Répartition de la difficulté :

- Niveau 1 : problème facile, ne nécessite pas de code complexe pour être mis en œuvre ;
- Niveau 2 : problème moyen, demande une réflexion plus poussée sur la présentation du code ;
- Niveau 3 : problème délicat, demande une réflexion poussée sur le problème en soi.
- Niveau 4 : problème difficile.

## 1 Aidez CPUMons !!! (niveau 2)

Le Kot associatif CPUMons a besoin de votre aide !

La liste des membres commence à s'allonger et il serait judicieux d'établir un système de gestion des membres avant que cette liste ne soit trop grande. Pour cela nous avons besoin :

- D'un objet "Membre" qui se caractérise par : un nom, un prénom, une adresse email UMons, une faculté, une section, une date d'inscription (sous la forme d'un nombre aammjj), son nombre de participation, ainsi que les dates auxquelles il a participé à un entraînement (aammjj).
- L'adresse email doit être construite sur base du prénom et du nom de l'étudiant (prénom.nom@student.umons.ac.be).
- On a besoin de connaître quels membres étaient présents à une date donnée ainsi que le nombre de membres présents à cette séance.
- Pour établir un graphique du nombre de membres présents à chaque date, on a besoin de connaître le nombre d'étudiants pour toutes les séances.
- Les membres seront stockés dans un tableau.
- On doit pouvoir ajouter et supprimer des membres du tableau.

## 2 Classement en moyenne (niveau 1.5)

Le but de cet exercice est de diviser une liste en deux sous-listes par rapport à la moyenne des éléments de la liste de base

**Exemple :** Soit  $L$  une liste : 2,4,95,7,32,-4,66,26,13, on peut en calculer la moyenne qui est égale à 26.78 et donc, les deux listes en sorties seront  $L_1$  : 2,4,7,-4,26,13 et  $L_2$  : 95,32,66.

**Remarques/Restrictions :**

- La complexité doit être, au maximum, en  $O(n^2)$  ;
- Vous ne pouvez créer qu'une nouvelle liste et non deux ;
- Si un élément est égal à la moyenne, il doit apparaître dans les deux listes.

**Challenge (niveau 3) :** Votre algorithme doit être en  $O(n \log(n))$ , mais il vous est autorisé d'utiliser les instructions du type  $t[2 : 5]$  (où  $t$  est une liste).

3 ...

4 ...

5 ...

## 6 Limiter le hasard... D'un jeu de hasard ! (niveau 4)

Vous avez été engagé par la Compagnie des Probabilités pour l'Utilisation Modérée mais Onéreuse de Nos Sous afin d'étudier un jeu de hasard relativement simple et paillard : le "cul de chouette"<sup>1</sup>. Ce jeu se joue avec un nombre  $N$  de dés à 6 faces ( $1 \leq N \leq 10$ ) que l'on lance une fois (les dés n'étant évidemment pas truqués. De plus, l'important, dans ce jeu, ce sont les valeurs : en l'occurrence la somme des résultats des  $N$  dés.

Votre tâche est d'écrire un algorithme donnant le nombre de manières possibles d'obtenir un résultat  $M$  ( $N \leq M \leq 6N$ ). Pour ce faire, il vous est conseillé d'effectuer les calculs pour  $N = 2$  et  $3$  afin de pouvoir généraliser la méthode.

**Remarque :** au vu de la faible valeur de  $N$ , il n'est pas nécessaire de se soucier de la complexité de l'algorithme créé.

## 7 Concours

Pour ceux qui veulent s'entraîner pour participer aux concours, voici quelques liens pour vous permettre de travailler sur ce qui vous intéresse à votre propre rythme :

1. [FranceIOI](#) : nécessite de débloquer les premiers niveaux, mais recouvre une très grande variété d'algorithmes et de problèmes différents rangés par thématique ;
2. [Isograd](#) : sur ce site, vous retrouverez de nombreux anciens concours dont, en particulier, les BattleDev précédentes qui constituent un *excellent* point de départ dans le monde des concours ;
3. [Google Code Jam](#) : les énoncés des années précédentes y sont disponibles. Le niveau requis est, bien évidemment, progressif et les problèmes sont en général assez intéressants ;
4. [Kattis](#) : site reprenant de *très* nombreux problèmes. Il vous est conseillé de tester un ou deux problèmes "triviaux" afin de bien vérifier si vous n'avez pas de problèmes avec les entrées et sorties ; puis, de passer aux faciles et, rapidement, aux moyens (les difficiles portant très bien leur nom).

---

1. Nom tout aussi paillard