

# DS d'informatique n°1 : ALGORITHMES DE PRONOTE

Durée : **2h**. Calculatrices non autorisées.

Le soin et la clarté pourront faire varier la note de  $\pm 1$  point. Soyez vigilants à respecter l'indentation, les notations de l'énoncé, et à mettre des commentaires pour les étapes complexes. **Vous pouvez réutiliser les fonctions des questions précédentes, même si vous n'avez pas su y répondre.**

L'outil Pronote a été une révolution dans les établissements d'enseignement public. Les élèves et leurs familles ont maintenant accès à leurs notes à tout moment ainsi qu'aux calculs automatisés des moyennes, des rangs et plus encore. Sur le principe, ces algorithmes semblent simples. Voyons un peu cela...

## Partie A : Notes des élèves sur un devoir

On suppose que les élèves ont composé un devoir et ont chacun une note. On dispose ainsi d'une liste qui contient toutes les notes (une par élève), typiquement :

[ 14, 3.75, 12.5, 7.25, ... ]

On s'intéresse tout d'abord à la moyenne et l'écart-type de la classe. Pour rappel, étant donné  $n$  nombres  $x_1, \dots, x_n$ , leur écart-type est donné par

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad \text{avec } \bar{x} \text{ la moyenne des nombres } x_1, \dots, x_n$$

- 1) Écrire une fonction `moyenne` qui à une liste de notes renvoie la valeur moyenne de cette liste.
- 2) Quel est l'écart-type de la liste [ 8, 11, 11 ] ?
- 3) Un informaticien en herbe a écrit la fonction `ecartType` suivante, qui à une liste de notes renvoie l'écart-type de cette liste. Cependant, il a fait une ou plusieurs erreurs. Recopier la fonction corrigée en soulignant chaque endroit que vous avez modifié.

```
1 def ecartType(L):
2     n = len(L)
3     M = Moyenne(L)
4     L2 = [ ( L(k) - M )**2 for k in range(L) ]
5     S = sum(L) / n
6     return S**1/2
```

Avant de rendre les copies, le professeur doit savoir quelle est la meilleure note, voire les trois meilleures.

- 4) Écrire une fonction `major` qui à une liste de notes renvoie la meilleure note.
- 5) Écrire une fonction `podium` qui à une liste de notes (ayant au moins trois notes) renvoie une liste contenant les 3 meilleures notes. *Indication* : on pourra utiliser l'instruction `L.remove(a)`, qui retire la première occurrence de la valeur `a` dans la liste `L` (mais produit une erreur si `a` n'est pas dans `L`).

On souhaite enfin vérifier que toutes les notes sur un même devoir sont "valides". Une note  $x$  est valide si  $x \in [0, 20]$  et si  $x$  est ou bien un entier, ou bien admet comme partie décimale 0.25, 0.5 ou 0.75.

- 6) On suppose que `x` est un flottant. Expliquer ce que fait l'instruction `int(x)`.
- 7) Écrire une fonction `validation` qui à un flottant `x` renvoie un booléen selon que la note `x` soit valide ou non.

## Partie B : Moyenne d'un élève sur une matière

On regarde maintenant, dans une matière donnée, toutes les notes obtenues par un(e) unique élève. Ces notes sont stockées dans une liste `M`. On cherche à calculer la moyenne de l'élève pour cette matière. Cependant, toutes les notes n'ont pas le même coefficient. On dispose ainsi d'une liste `Coeff` qui a la même taille que la liste `M` : pour chaque indice `i`, la note `M[i]` est pondérée par le coefficient `Coeff[i]`.

8) Déterminer la moyenne pondérée de l'élève lorsque

`M = [ 12, 16, 6 ]`      `Coeff = [ 3, 1, 4 ]`

9) Écrire une fonction `moyMatiere` qui à une liste de notes `M` retourne la moyenne pondérée de l'élève.

10) Quel problème présente la fonction `moyMatiere` telle qu'elle est implémentée ? On discutera en termes de variable locale et de variable globale.

On considère à présent que l'élève a pu avoir une ou plusieurs absences. En cas d'absence au devoir d'indice `i`, alors on note `M[i]="abs"` : la note de ce devoir ainsi que son coefficient ne sont pas pris en compte pour le calcul de la moyenne de l'élève. Par exemple, si

`M = [ 12, 16, "abs" ]`      `Coeff = [ 3, 1, 4 ]`

alors la moyenne de l'élève est 13.

11) Écrire une fonction `moyMatiere2` similaire à `moyMatiere` et qui tient compte des absences.

Le professeur a également fait 10 devoirs maison pendant l'année. Ces devoirs sont notés par des lettres majuscules, allant de `A` à `E`. Ainsi, pour chaque élève, on dispose également d'une chaîne de caractères `DM` de taille 10 qui contient les lettres associés à ces devoirs.

12) Écrire une fonction `comptage` qui à une chaîne de caractères `S` et à un caractère `c` retourne le nombre d'occurrences de `c` dans `S`.

Un bon professeur cherche à encourager les élèves qui fournissent un travail sérieux et régulier. Un élève est un "chouchou" du professeur s'il a obtenu au moins six `A` sur les dix devoirs maison. Dans ce cas, sa moyenne est augmentée de 10% (sans pour autant dépasser 20).

13) Écrire une fonction `moyFinal` qui à une liste de notes `M` et une liste de lettres `DM` retourne la moyenne.

## Partie C : rang d'un élève sur une matière

On suppose qu'on a stocké les moyennes des élèves pour une matière donnée dans une liste `MOY`. De plus, on assigne à chaque élève un indice : `MOY[i]` est ainsi la moyenne de l'élève d'indice `i`.

14) Écrire une fonction `position(L,a)` qui retourne la position de la première occurrence de la valeur `a` dans la liste `L`.

On suppose qu'on dispose d'une fonction `tri` qui prend en argument une liste `L` et retourne une liste de même taille avec les valeurs triées par ordre croissant.

15) Écrire une fonction `mediane(MOY)` qui retourne la médiane des notes contenues dans `MOY`, ainsi que les notes inférieures à cette médiane. Rappel : la médiane est le plus petit réel `m` tel qu'au moins 50% des notes soient inférieures à `m`.

16) Écrire une fonction `rang(MOY,i)` qui retourne le rang de l'élève numéro `i` dans la liste des moyennes `MOY`. En cas d'ex-aequo, on retournera toujours le rang le plus bas. Par exemple les instructions `rang( [ 14, 11, 11 ] , 2 )` et `rang( [ 14, 11, 11 ] , 1 )` retourneront 3.

Il y a 10 types de personnes : celles qui écrivent les nombres en binaire, et les autres !