

# Prologin 2014

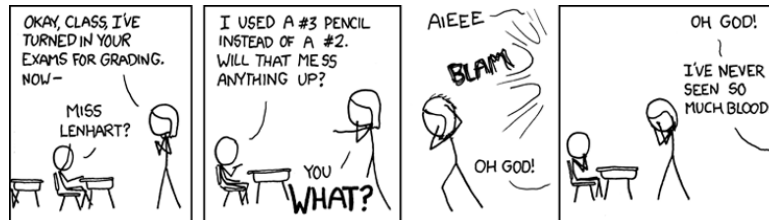
Concours national d'informatique

**Épreuve écrite d'algorithmique**  
**Lille & Toulouse**

Samedi 22 février 2014



# YOUR TURN TO PICK



XKCD – « Scantron », par Randall Munroe

## 1 Préambule

Bienvenue à **Prologin**. Ce sujet est l'épreuve écrite d'algorithmique et constitue la première des trois parties de votre épreuve régionale. Sa durée est de 3 heures. Par la suite, vous passerez un entretien (20 minutes) et une épreuve de programmation sur machine (4 heures).

### Conseils

- Lisez bien tout le sujet avant de commencer.
- **Soignez la présentation** de votre copie.
- N'hésitez pas à poser des questions.
- Si vous avez fini en avance, relisez bien, ou préparez votre présentation pour l'entretien.
- N'oubliez pas de passer une bonne journée.

### Remarques

- Le barème est donné à titre indicatif uniquement.
- Indiquez lisiblement vos nom et prénom, la ville où vous passez l'épreuve et la date en haut de votre copie.
- Tous les langages sont autorisés, veuillez néanmoins préciser celui que vous utilisez.
- Ce sont des humains qui lisent vos copies : laissez une marge, aérez votre code, ajoutez des commentaires (**seulement** lorsqu'ils sont nécessaires) et évitez au maximum les fautes d'orthographe, sinon ça va barder.
- Le barème récompense les algorithmes les plus efficaces : écrivez des fonctions qui trouvent la solution le plus rapidement possible.
- Si vous trouvez le sujet trop simple, relisez-le, réfléchissez bien, puis dites-le-nous, nous pouvons ajouter des questions plus difficiles.

## 2 Sujet

### Introduction

Chaque année, de pauvres organisateurs Prologin doivent corriger des centaines de copies, celles-là mêmes que vous vous apprêtez à remplir. Vous pouvez dès maintenant abandonner et rendre feuille blanche, l'équipe de correction vous remerciera<sup>1</sup>. Mais vous devriez pouvoir nous aider d'une autre manière<sup>4</sup>.

Après la correction de l'épreuve machine et de l'épreuve écrite, les organisateurs passent de très longues heures à sélectionner les finalistes. Le gros problème qu'ont les organisateurs, c'est que tous les candidats ont trois notes (écrit, machine, entretien). Par exemple, sur la figure 1, bien que Clément ait bien réussi son épreuve machine, Jérémie a obtenu une meilleure note à l'écrit, ces deux candidats sont donc *incomparables*.

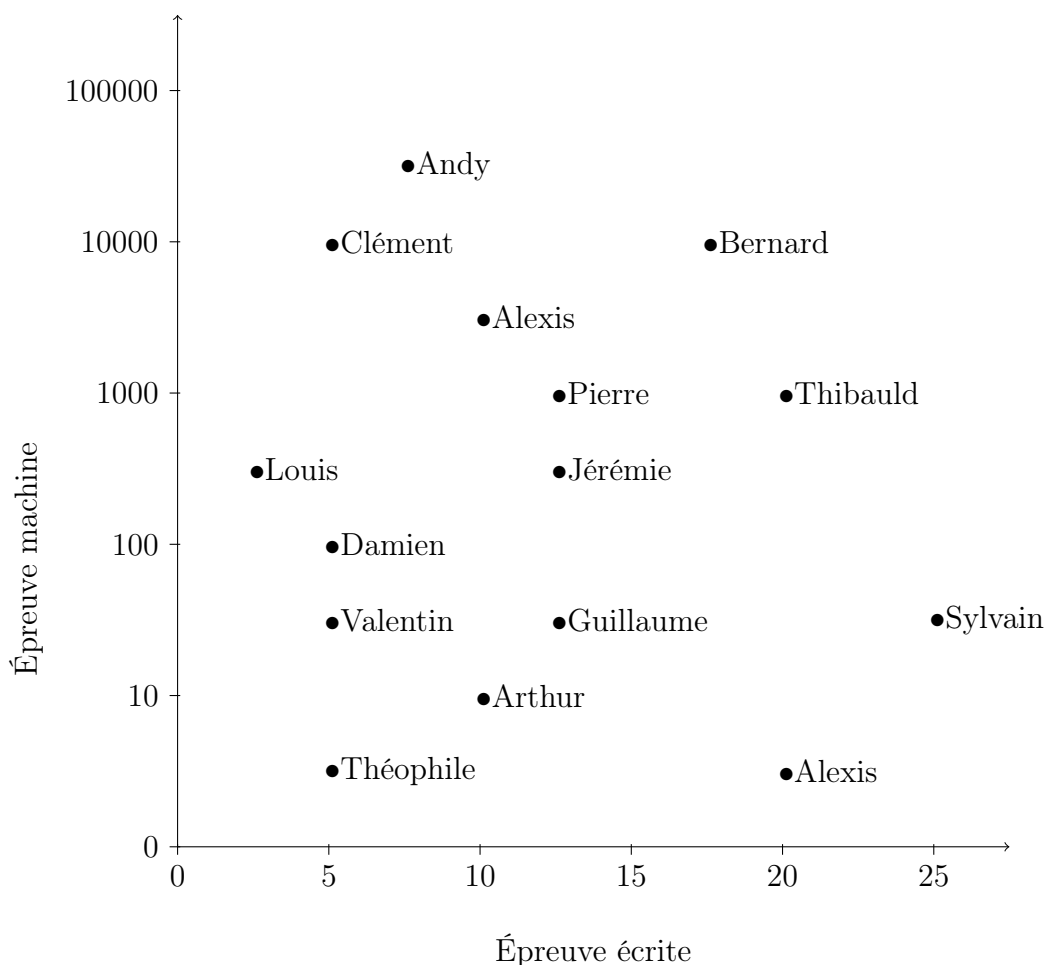


FIGURE 1 – Les notes de 15 candidats sur l'épreuves de Toulouse.

Par concision, on ignorera la note d'entretien dans la suite du sujet.

Plusieurs fonctions prennent en argument une liste de notes, il s'agit des notes aux deux épreuves, vous devez donc prendre en argument une liste de paires.

1. Vous pouvez aussi abandonner après avoir lu le sujet<sup>3</sup>.

3. Le fait que l'auteur du sujet pense à ce genre de chose en écrivant l'introduction est un mauvais signe.

4. Ouf.

## Question 1

(2 points)

Une première approche pour trouver les meilleurs candidats est d'utiliser une moyenne.

Écrire une fonction qui prend en argument la liste des notes aux deux épreuves, calcule pour chaque candidat sa moyenne et renvoie la meilleure moyenne.

On dit qu'un candidat A *domine* un candidat B lorsque A a<sup>5</sup> une meilleure note que B à l'écrit ou en machine et n'a pas de note plus mauvaise que B sur l'autre épreuve, c'est à dire :  $\text{écrit}_A > \text{écrit}_B$  et  $\text{machine}_A \geq \text{machine}_B$  ; ou  $\text{écrit}_A \geq \text{écrit}_B$  et  $\text{machine}_A > \text{machine}_B$ .

Par exemple, sur la figure 1, Jérémie domine Théophile et Pierre domine Jérémie.

## Question 2

(2 points)

Donner le nom des candidats *imbattus* sur l'exemple de la figure 1, c'est-à-dire les candidats qui ne sont dominés par aucun autre.

## Question 3

(3 points)

Écrire une fonction qui prend en argument vos notes (une paire (écrit, machine)) et une liste des notes des autres candidats aux deux épreuves et renvoie si vous êtes un candidat imbattu.<sup>6</sup>

## Question 4

(2 points)

Pour rendre le calcul de la moyenne plus équitable, il est proposé d'introduire des coefficients de pondération  $C_e$  et  $C_m$ , ainsi la moyenne d'un candidat n'est plus  $\frac{\text{écrit} + \text{machine}}{2}$  mais  $\frac{C_e \times \text{écrit} + C_m \times \text{machine}}{C_e + C_m}$ .

Malheureusement, il semblerait qu'il est possible pour un candidat d'être imbattu mais de ne jamais être premier avec un calcul de moyenne quel que soient les coefficients de pondération utilisés.

Donner un exemple de notes (comme il est fait sur la figure 1) sur lequel un candidat imbattu ne peut avoir la meilleure moyenne pondérée.

La moyenne pondérée ne permettant pas de classer correctement les candidats, on s'intéresse maintenant aux relations « dominé-dominant ». On représente une relation de dominance par un couple  $(A, B)$  signifiant que A domine B. On veut utiliser une liste des dominances à la place de la liste des notes. Ainsi, à la place des notes de la figure 1, on aura une liste contenant le couple (Thibault, Valentin) mais pas le couple (Arthur, Pierre).

## Question 5

(3 points)

Écrire une fonction qui prend en argument la liste des notes aux deux épreuves et renvoie une liste des dominances comme décrite ci-dessus.

---

5. Ah.

6. En plus d'être considérée fautive, la réponse `return true` sera récompensée de points négatifs.

## Question 6

(4 points)

Écrire une fonction qui prend en argument la liste des dominances et renvoie la liste des candidats imbattus.

## Question 7

(5 points)

Le problème de cette liste, c'est qu'il y a beaucoup de relations qui ne servent à rien. Par exemple, sur la figure 1, la relation « Guillaume domine Théophile » est inutile, sachant qu'on peut la déduire de « Guillaume domine Arthur » et « Arthur domine Théophile ».

Écrire une fonction qui prend en argument une liste des dominances et renvoie la liste des dominances équivalentes la plus petite possible.

## Question 8

( $2,5 \times 2^2$  points)

Certains organisateurs pensent que les candidats avec de bonnes notes à l'écrit ont aussi de bonnes notes en machine. Vous allez écrire de quoi nous épargner une longue explication.

Écrire une fonction qui prend en argument la liste des notes aux deux épreuves et renvoie la plus longue liste de notes  $(e_1, m_1), (e_2, m_2), \dots, (e_n, m_n)$  tel que les  $e_i$  sont triés dans l'ordre croissant et les  $m_i$  sont triés dans l'ordre décroissant.

Vous pouvez attaquer les questions suivantes si et seulement si vous vous êtes attentivement relu aux questions précédentes et si votre copie est blanche.

## Question bonus 9

(1 point)

À quoi correspond l'ensemble des candidats pouvant être considérés comme meilleurs par moyenne pondérée ?

## Question bonus 10

(7 points)

Les organisateurs ont surtout des problèmes quand il y a beaucoup de candidats incomparables. Écrivez une fonction qui retourne la taille du plus grand ensemble de candidats qui ne peuvent être comparés entre eux.

Le sujet est sur 31 points, et les questions bonus rapportent au total 8 points, plus 1 point de présentation.

---

2. Si votre algorithme est en  $O(n \log(n))$  ceci est un carré, sinon, c'est une note en bas de page.