

1/ REMARQUES GÉNÉRALES

Le sujet était composé de deux parties indépendantes.

Il abordait des notions du programme de première année (algorithmes simples, manipulation de listes et de chaînes de caractères, terminaison et complexité d'un algorithme, représentation des nombres, bases de données) et de deuxième année (récursivité et tri).

Les correcteurs constatent des lacunes liées à la notion de listes (création d'une liste, utilisation de *append*, utilisation d'une boucle *for* pour parcourir les éléments d'une liste) et de chaînes de caractères (non « mutables »).

Bien que souvent traitées, les questions liées aux bases de données manquent régulièrement de rigueur dans la syntaxe.

De nombreux problèmes de programmation sont récurrents :

- problème d'indentation ;
- utilisation de « $L[i]=x$ » pour une liste vide ;
- utilisation de « $L=L+x$ » au lieu de « $L=L+[x]$ » ;
- utilisation de « $L=L.append(x)$ » ;
- problème d'indices lors de parcours de listes ;
- oubli de « return » ;
- l'utilisation de chaînes de caractères pose de nombreux problèmes (instructions licites ou non, caractère « non mutable »...).

Certaines copies sont peu lisibles et non soignées. Les correcteurs ont pris cela en compte dans l'évaluation. Il est vivement conseillé de barrer proprement en cas d'erreur et de se relire (fautes d'orthographe).

2/ REMARQUES SPÉCIFIQUES

Q1. Les justifications données sont décevantes : indiquer « ce n'est pas optimal car il reste de la place » n'est pas suffisant. De nombreux candidats ne font pas attention à l'énoncé et donnent des réponses complètement fausses (plusieurs fois le même produit dans le camion, utilisation de plusieurs camions...).

Q2. Généralement bien traitée.

Q3. Généralement bien traitée.

Q4. Question souvent mal traitée. La création d'une liste pose de nombreux problèmes et même dans le cas d'une liste bien créée, les indices sont souvent faux.

Q5. De nombreux problèmes d'indices.

- Q6.** Le nombre d'itérations donné est souvent faux. Certains candidats donnent des valeurs entières pour L.
- Q7.** Le caractère « non mutable » d'une chaîne de caractères est peu connu.
- Q8.** Bien que le tri d'insertion soit souvent reconnu, il semble que de nombreux candidats répondent au hasard. Rappelons que le programme ne contient que trois tris. Les complexités demandées sont souvent connues mais des réponses absurdes se retrouvent dans les copies.
- Q9.** De nombreuses méthodes données pour cette question mais aussi souvent des problèmes de syntaxe. Cette question peut être considérée comme un raisonnement classique. Un nombre non négligeable de candidats veulent répondre à la question à l'aide de tris.
- Q10.** De faux arguments de complexité souvent donnés pour cette question.
- Q11.** Très bien traitée par un petit nombre de candidats.
- Q12.** Sv est souvent renvoyé mais la condition du while est rarement correcte.
- Q13.** Correctement traitée mais de nombreux candidats pensent que $4/3$ est supérieur à $3/2$.
- Q14.** Laborieux. Travailler le stop d'un appel récursif, en quelques mots, est une compétence attendue.
- Q15.** Très peu abordée. Certaines réponses sont complètement hors-sujet (« il faut enlever des cartons »...).
- Q16.** Question simple mais certains candidats se trompent en utilisant un while au lieu d'un if.
- Q17.** Question non triviale et globalement réussie.
- Q18.** Question pas forcément bien comprise.
- Q19.** Syntaxe rarement parfaite.
- Q20.** Question difficile, assez bien abordée par les candidats.
- Q21.** Complètement ratée. La notion de clé primaire n'est pas du tout acquise par une large majorité de candidats.
- Q22.** Requête très simple et pourtant de nombreux candidats ne connaissent pas la syntaxe associée au SQL.
- Q23.** De même que Q22. Plusieurs candidats souhaitent utiliser « SELECT date AND heure ».
- Q24.** De même que Q22.
- Q25.** Question simple, très bien réussie.
- Q26.** La faute de frappe (« est » apparaissait deux fois) n'était évidemment pas la bonne réponse. Beaucoup de candidats pensent que la fonction « len » ne peut pas s'utiliser sur une chaîne de caractères. De même, l'instruction « Bin[i] » ne semble pas licite dans de nombreuses copies.
- Q27.** Très peu de bonnes réponses pour cet algorithme que l'on peut considérer comme classique.
- Q28.** Correctement traitée mais parfois un certain manque de rigueur (à 1 bit près).