

EPREUVE D'INFORMATIQUE

Durée : 3 heures

PRESENTATION DU SUJET

Le sujet consiste en une série d'exercices basés sur le thème du tri permettant de vérifier que les candidats maîtrisent le programme de l'option informatique. Les 2 premières parties font appel à des notions de base. Les parties 3 et 4 sont moins faciles ; elles demandent au candidat de faire preuve de d'aptitude en recherche et en analyse.

COMMENTAIRE GENERAL DE L'EPREUVE

Le problème du langage utilisé reste entier, bien qu'une forte majorité de candidats ait choisi de composer en Caml (75 %). Les autres langages utilisés ont été Pascal (24 %) et une infime minorité a composé en Maple. Il faudrait absolument prendre des mesures pour unifier les compositions (remarques déjà faites dans les rapports 2002, 2003 et 2004). Imagine-t-on des copies de mathématiques ou de physique rédigées en allemand, français et espagnol ?

La discrimination entre les compositions s'est surtout faite sur les points suivants :

- une mauvaise lecture de l'énoncé pénalise également de nombreux candidats ;
- le "Pourquoi faire simple quand on peut faire compliqué ?" s'est encore appliqué souvent (*des candidats ont exprimé un résultat n sous la forme $2^{\log(n)/\log(2)}$*) ;
- les exercices inhabituels (*heap sort*) sont délaissés alors que les exercices familiers (*recherche dichotomique*) sont traités à la va-vite ;
- l'absence de commentaires, la programmation obscure d'exercices, la mauvaise présentation et l'écriture parfois illisible rend certaines copies difficiles à lire et donc à corriger.

En résumé, encore une fois, les causes d'une mauvaise composition ont été plus souvent une mauvaise intelligence des connaissances acquises dans les années de préparation aux concours qu'une absence de ces connaissances.

ANALYSE PAR PARTIES

1. Le tri par sélection

Cette partie ne présentait pas de difficulté spéciale, l'essentiel étant de connaître la différence entre *récursif* et *itératif*. Plusieurs candidats ont commis la faute de chercher la valeur du plus grand élément (algorithme sans doute traité en classe), puis faire un deuxième parcours pour trouver l'indice. Le calcul de complexité a été en général bien traité, mais ici comme dans le deuxième exercice, certains candidats considèrent que la somme des n premiers nombres est égale à $n!$

2. Tri par insertion

La différence entre les copies s'est faite ici essentiellement sur trois points.

- dans la recherche de la position d'insertion, beaucoup de candidats ont négligé de tester la limite du tableau (*nombre supérieur à tous les éléments du tableau*) ;
- le décalage des éléments d'une position vers la fin dans le tableau s'est fait souvent en écrasant les valeurs à décaler (*boucle dans le sens montant, sans sauvegarde*) ;
- dans ce même exercice, certains candidats ont utilisé un tableau auxiliaire, ce qui était évidemment pénalisant.

3. Tri rapide

C'était la partie la plus difficile. Les deux premiers exercices étaient pourtant assez simples, mais beaucoup les ont délaissés pour passer directement à la partie 4 ; parmi ceux qui ont traité complètement cette partie, aucun candidat n'a écrit une fonction de partition correcte.

4. Heap sort

Bien que conceptuellement plus difficile, cette partie était davantage guidée par l'énoncé, ce qui a permis à beaucoup de candidats de gagner des points sur les questions simples de début.

ANALYSE DES RESULTATS

Le niveau général est correct, les candidats possédant, à quelques exceptions près, la compréhension des structures de fonctionnement d'un programme informatique. Le classement des copies ressemble à une course par élimination : il y a plus d'exercices de difficulté moyenne mal traités que d'exercices difficiles correctement traités.

CONSEILS AUX FUTURS CANDIDATS

- Lisez bien tout l'énoncé.
- Planifiez les durées que vous allez consacrer à chaque exercice et tenez-y-vous.
- Repérez les exercices que vous pensez pouvoir traiter sans difficulté.
- Faites un brouillon et rédigez proprement votre copie.
- Étudiez bien le contexte de l'énoncé, faites tourner vos idées de programmes *à la main*.
- Commentez vos programmes.
- Choisissez des noms parlants pour vos identificateurs.
- Mettez en page vos programmes en utilisant un style d'indentation clair.
- Écrivez lisiblement