EPREUVE D'INFORMATIQUE

Durée : 3 heures

PRESENTATION DU SUJET

L'épreuve se compose de six exercices indépendants :

Exercice 1 : Le candidat doit proposer trois programmes : un pour tester le caractère symétrique d'une matrice carrée, un pour tester l'éventuelle égalité de deux listes et un pour fusionner deux tableaux dont les éléments sont triés dans le même ordre, après avoir vérifié cette propriété.

Exercice 2 : Il s'agit de lire des programmes écrits dans un pseudo-langage et de déterminer ce que ces programmes calculent. Le premier programme est une simple boucle, le second plus complexe détermine la ℓ-ième valeur d'un tableau dans l'ordre croissant.

Exercice 3 : Le candidat doit proposer un programme permettant d'établir la liste des entiers naturels parfaits (égaux à la somme de leurs diviseurs stricts) inférieurs ou égaux à 9999.

Exercice 4 : Le candidat doit proposer un programme permettant de calculer le nombre de fois nécessaire pour passer d'un entier naturel à un unique chiffre en multipliant ses chiffres successivement.

Exercice 5 : Il s'agit de calculer des expressions rationnelles pour les langages reconnus par une famille paramétrée d'automates.

Exercice 6 : C'est un exercice de logique dans lequel il fallait modéliser des contraintes vérifiées par une flotte d'avions sous forme logique, puis écrire plusieurs tables de vérités et les interpréter.

COMMENTAIRE GENERAL DE L'EPREUVE

L'épreuve est conçue pour vérifier les connaissances des candidats sur divers aspects du programme : les quatre premiers exercices sont des exercices de programmation, les deux suivants sont plus théoriques, un sur la théorie des automates et un de logique. 296 candidats ont composé cette épreuve. Les notes se sont étalées entre 1 à 19, avec une moyenne de 9,99 et un écart-type de 3,79. La plupart des candidats proposent leurs programmes en langage camL.

ANALYSE GENERALE

Dans la plupart des copies, tous les exercices sont abordés, le troisième et le dernier s'avérant les plus faciles. Certains semblent maitriser tous les sujets mais dans l'ensemble, la rédaction souffre d'absences de justification et de commentaires. En ce qui concerne les programmes, rappelons que les correcteurs n'attachent pas une importance démesurée à la correction de la syntaxe, mais plutôt à l'articulation du programme, l'ordre dans lequel les variables sont affectées, la justesse des tests logiques nécessaires, la terminaison du programme... et au fait qu'avec d'éventuelles petites corrections syntaxiques, il donnerait le bon résultat. Il en apprécie aussi l'efficacité du point de vue de la complexité. En ce qui concerne les exercices plus théoriques, les correcteurs apprécient la précision de l'argumentation.

ANALYSE DES RESULTATS PAR PARTIES

- Le premier exercice est un exercice de programmation.

- Ecrire un programme pour tester si une matrice carrée est symétrique ou non. L'énoncé demandait que la sortie du programme soit un booléen ; l'erreur la plus fréquente est de lui affecter une valeur à chaque test (sur les coefficients de la matrice), ce qui bien sûr, rend une réponse erronée. Les correcteurs ont apprécié l'efficacité du programme (pas de tests inutiles ou de tests effectués deux fois).

- Ecrire un programme pour tester l'éventuelle égalité de deux listes Cet exercice est plutôt bien réussi dans l'ensemble. L'erreur la plus fréquente est d'introduire une dissymétrie entre les deux listes qui aboutit finalement à ne tester qu'une éventuelle inclusion.

- Ecrire un programme qui peut réaliser la fusion de deux tableaux triés dans le même sens. Ce programme s'est avéré beaucoup plus difficile pour les candidats. Un nombre non négligeable de candidats l'ont abandonné en cours de route. L'énoncé proposait d'écrire un premier programme déterminant si un tableau est trié dans l'ordre croissant, décroissant, constant ou non trié. De nombreux candidats ont interprété le fait de ne pas être constant pour un tableau trié par le fait d'être strictement croissant ou strictement décroissant, ce qui, bien sûr, ne convient pas. L'idée était de séparer le cas constant des cas croissants et décroissants, car un tableau constant peut ensuite être fusionné et à un tableau croissant, et à un tableau décroissant. Peu de candidats s'en sont aperçus et ont proposé une solution complètement juste. Souvent, seuls les tableaux tries non constants étaient pris en compte. Enfin, signalons la solution originale qui consiste à mettre les deux tableaux bout à bout, s'ils peuvent être fusionnés, puis à trier le tableau final.

- Le deuxième exercice est une lecture de programmes écrits en pseudo-langage. Si de nombreux candidats peuvent donner le bon résultat, très peu justifient leur analyse (le rôle des différentes variables, le découpage logique du programme ne sont quasiment jamais mentionnés). Pourtant, pour un programme un peu complexe, le candidat s'est probablement posé toutes ces questions pour arriver au résultat.

- Le premier est une simple boucle et il fallait comprendre l'usage du tant que, ce qui n'est pas toujours acquis. Une réponse informelle du type "ce programme calcule la partie entière de $\sqrt{2006}$ plus 1 était acceptée, on n'exigeait pas la valeur 45.

- Le deuxième programme calcule de façon récursive la ℓ-ième valeur d'un tableau, dans l'ordre croissant. Ce programme est beaucoup plus difficile et a été moins souvent abordé. Des candidats ont eu des difficultés avec le programme intermédiaire qui tronquait le tableau initial et donc avait pour résultat un tableau. La très grande majorité des candidats qui ont tenté de lire le programme ont simplement énoncé des résultats, très peu ont su l'argumenter, voire le justifier de façon précise.

- Le troisième exercice sur les nombres parfaits a été bien réussi. Le programme doit faire plusieurs tâches : tester si un nombre est parfait (en parcourant la liste de ses diviseurs) puis faire la liste demander. Le jury a apprécié que les diverses tâches soient séparées ou au moins commentées.

- Le quatrième exercice consiste en un programme pour calculer le plus petit entier de persistance égale à 5, la persistance d'un entier étant le nombre de fois où on multiplie les chiffres d'un entier naturel pour arriver à un seul chiffre. Là encore, le programme devait effectuer plusieurs tâches (produit des chiffres d'un entier, persistance d'un entier, puis la recherche du plus petit de persistance 5 (679))et les correcteurs ont apprécié la clarté du découpage. Signalons que la persistance n'étant pas une fonction croissante, il n'est pas évident que la persistance reste < 5 avant d'atteindre 5 (c'est en fait le cas) ; le programme devait a priori en tenir compte.

- Le cinquième exercice, sur les expressions rationnelles des langages reconnus par certains automates, s'est avéré difficile, encore une fois par manque de techniques d'analyse: de nombreux candidats semblent être capables de deviner le langage reconnu par un automate très simple, mais ne sachant pas comment le justifier, ils se trouvent déroutés dés que l'automate est un peu plus complexe.

- Le sixième exercice de logique était très facile et a été traité correctement et précisément justifié dans un grand nombre de copies.

CONSEIL AUX FUTURS CANDIDATS

- Il n'est bien sûr pas facile d'écrire sur papier un programme. Le correcteur conseille donc aux candidats de clairement commenter leurs programmes, afin d’éviter certaines confusions. Si le programme s'avère long, il est raisonnable de le découper en divers sous-programmes dont on commente la finalité. On peut lors de l'introduction d'une variable écrire en commentaire le rôle qu'elle va jouer dans le programme et justifier son initialisation.

- Pour la lecture d'un programme, on peut aussi expliquer le rôle de chaque variable et la finalité des sous-programmes. En cas d'hésitation, on peut bien entendu tester le programme sur des données très simples.

- Les exercices plus théoriques doivent être argumentés précisément, tout comme un exercice de mathématiques.