

Composition d'Informatique (2 heures), Filières MP et PC

Rapport de MM. Olivier LY et Dominique ROSSIN, correcteurs.

Les statistiques et le bilan donnés ci-après concernent uniquement les candidats admissibles français.

I. Bilan

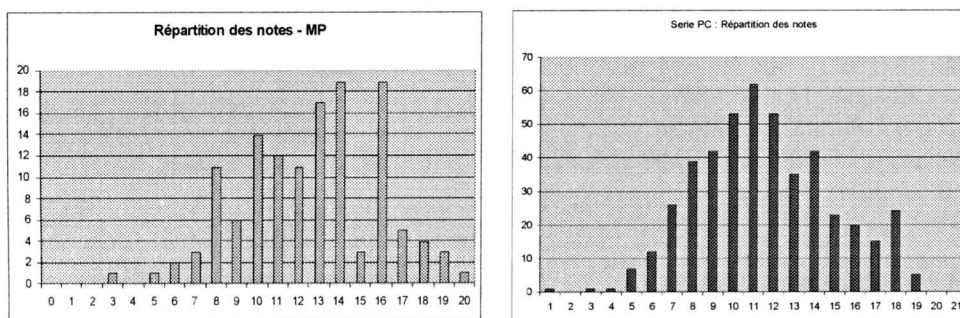
La répartition des notes pour les 456 candidats français admissibles PC et des 132 candidats MP est la suivante :

Note	$N < 5$	$5 \leq N < 9$	$9 \leq N < 13$	$13 \leq N < 17$	$17 \leq N$
MP	2%	16%	41 %	35%	6%
PC	2%	26%	44%	22%	6%

Série PC : La moyenne des copies s'élève à 11,06 avec un écart-type de 3,36.

Série MP : La moyenne des copies s'élève à 12 avec un écart-type de 3,4.

Une analyse plus fine des notes donne la distribution suivante :



Le langage de programmation utilisé est très majoritairement Maple (que trop de candidats continuent à orthographier Mapple). Néanmoins, certains candidats ont choisi Java, Mathematica ou encore Pascal pour développer leurs solutions. A noter également qu'un candidat a utilisé Ocaml et Scilab.

II. Commentaires

Certaines questions du sujet demandaient l'écriture d'un programme. Le critère principal d'évaluation pour ces questions est la conformité du programme aux spécifications imposées par l'énoncé, que ce soit du point de vue de la forme que du fond. Néanmoins ont été pris en compte les algorithmes employés, la clarté et modularité des programmes, et enfin la complexité en temps et en espace. Remarquons que dans certains cas très rares, le style a été également pris en compte : écrire l'ensemble d'un programme sur une seule ligne n'est pas acceptable.

Dans l'ensemble, les candidats font peu d'erreurs de syntaxe. La plus grande partie des erreurs est due aux algorithmes employés ou encore plus généralement à des réponses erronées. Seul l'emploi pas toujours maîtrisé de la structure Maple

```
for <expr> from <expr> to <expr> while <expr> do <statement> od ;
```

bien que déjà fortement déconseillée les années précédentes a conduit certains candidats à faire des erreurs au niveau de la terminaison et de la complexité des algorithmes. Notons que la confusion entre affichage et valeur retour de fonction persiste encore chez certains candidats.

Certains langages de programmation ne permettent pas de modifier certains paramètres d'entrée. Bien entendu, cela a été pris en compte lors de la correction et l'intégralité des points a été accordée si les variables avaient la bonne valeur en fin de fonction.

Les autres questions du problème portaient essentiellement sur des propriétés combinatoires des objets mis en jeu dans le sujet. Pour l'évaluation de ces questions, il a été tenu compte des résultats et des raisonnements bien sûr, mais également de la clarté de l'énoncé des démonstrations.

Le problème a été traité jusqu'à la **question 11** incluse par une grande partie des candidats. Pour la filière PC, les **questions 12** et **13** n'ont été abordées que dans une proportion plus modeste en raison de leur difficulté ainsi que probablement de la longueur du sujet. Ces deux dernières questions ont été en revanche abordées par bon nombre de candidats MP ce qui explique le décalage des notes. Le barème était légèrement différent quant à ces deux dernières questions entre les MP et les PC.

III. Commentaires détaillés

Question 1

Il s'agissait ici de calculer le coût d'une séquence de déplacements pour un bloc de requêtes. La question a été traitée par la plupart des candidats. La majorité des programmes proposés retournaient un résultat correct. Mais souvent avec une complexité algorithmique mauvaise : Beaucoup de candidats ont donné des solutions comportant plusieurs passes sur la liste des déplacements, calculant les coûts liés à chaque tête séparément. Cette

erreur a été pénalisée. Remarquons que pour une solution optimale, une seule variable de position est nécessaire.

Question 2

Il s'agissait d'avancer l'argument selon lequel le problème est symétrique, et que l'on peut échanger les 2 têtes. La plupart des candidats l'ont remarqué. En revanche, la rédaction a souvent fait preuve de manque de clarté.

Question 3

La réponse était 2ⁿ. Cette question a été traitée correctement par quasiment tous les candidats.

Questions 4,5, et 6

Ces questions ont été traitées correctement par quasiment tous les candidats.

Question 7

Cette question a été correctement traitée par la plupart des candidats. Ont été pénalisés cependant les candidats ne donnant pas explicitement l'énumération des cas possibles comme il était demandé dans l'énoncé de la question.

Question 8

Il s'agissait dans cette question d'écrire un programme comparant les 4 solutions possibles. Beaucoup de candidats ont abordé cette question avec les bonnes idées. Néanmoins, la complexité relative de l'enchaînement des tests nécessaires à la tâche a conduit beaucoup de candidats à commettre des erreurs d'étourderie et de syntaxe qui ont été pénalisées. Les candidats ayant donné une solution correcte à cette question ont pour la plupart tiré profit d'un souci évident de clarté dans le style de l'écriture de leurs programmes. De nombreux candidats ont essayé d'éviter l'énumération des 4 solutions en proposant des séries de tests. Malheureusement, seulement un d'entre eux a donné une solution satisfaisante.

Question 9

Il s'agissait d'indiquer comment lire le résultat demandé dans le tableau coût. Parmi les solutions correctes calculant le minimum du tableau, les candidats ayant remarqué qu'il suffit de calculer le minimum de la dernière ligne et de la dernière colonne ont été privilégiés.

Question 10

Cette question comportait plusieurs propriétés du tableau coût à démontrer. Beaucoup

de candidats l'ont abordé. La plupart d'entre eux ont donné des démonstrations convaincantes pour les propriétés 1, 3 et 4. En revanche, la seconde propriété n'a été comprise que par une minorité de candidats ; et sa démonstration n'a été que rarement claire.

Question 11

Pour cette question, il s'agissait d'exploiter les propriétés vues en **question 10** et notamment la seconde pour obtenir un programme calculant le pas de récurrence élémentaire pour le calcul de coût. Cette question a été abordée par une majorité de candidats. Néanmoins, les programmes proposés comportaient beaucoup d'erreurs, tant sur le fond en raison d'une mauvaise compréhension de la seconde propriété de la **question 10**, que sur la forme, souvent dues à un manque de modularité dans l'écriture du programme.

Questions 12 et 13

Pour la **question 12**, il s'agissait d'exploiter les propriétés vues en **question 10** et la procédure donnée en **question 11** pour mettre en œuvre l'itération conduisant au calcul du tableau coût. Il s'agissait d'écrire l'itération avec soin sans omettre de définir correctement les conditions initiales.

La **question 13** portait sur une optimisation en espace de l'algorithme développé en **question 12**.