

4 – INFORMATIQUE

Epreuve écrite – filière MP

I) REMARQUES GENERALES

Généralités.

Le sujet est composé d'un problème qui permet de couvrir une large partie du programme. Il traite de l'étude de la recherche d'une sous-chaîne dans une chaîne de caractères. Plusieurs méthodes sont traitées dans ce problème. Plus précisément, il s'agit de préciser chaque occurrence d'une chaîne de caractères donnée dans une autre chaîne de caractères également donnée.

Ce problème aborde des éléments de la logique, de la théorie des langages et des automates et de la programmation. Il traite de chaînes de caractères représentées sous forme de tableaux. Ensuite, différents traitements, permettant de retrouver les occurrences d'une chaîne de caractères dans une autre, sont abordés. Ces traitements conduisent à des programmes itératifs et récursifs. Il est proposé de construire de manière progressive ces algorithmes. Un algorithme efficace est construit après plusieurs étapes.

Globalement, les candidats abordent les trois premières parties du problème.

La présentation ainsi que la rédaction sont bonnes dans l'ensemble. Plusieurs candidats omettent de faire des références explicites aux questions déjà traitées dans le sujet lors de la réponse à une question qui nécessite l'utilisation d'un résultat déjà établi.

Beaucoup de candidats omettent l'indentation de leurs programmes, ils produisent des programmes CAML ou PASCAL mal indentés. Cela a pour conséquence de rendre la lecture de ces programmes difficile.

Sur le plan de la programmation, on constate toujours une utilisation abusive des exceptions et de la construction *try*. Celle-ci est le plus souvent inappropriée, de simples instructions conditionnelles pouvaient éviter son utilisation. De manière générale, plusieurs candidats utilisent ce type de mécanisme pour interrompre une boucle *while* ou bien *for*.

Le traitement des questions relatives à la complexité des algorithmes n'a pas toujours été correctement abordé. Nous avons noté :

- que les questions portant sur la complexité n'ont pas été abordées ou que les calculs de complexité étaient souvent erronés et ne correspondent pas aux programmes décrits dans les copies ;
- que les programmes en réponses aux questions n'étaient pas conformes à la complexité exigée par la question. Par exemple, certaines questions exigent une complexité linéaire et les réponses apportées ne respectent pas cette complexité demandée.

Remarques particulières

Les remarques ci-dessous sont spécifiques aux questions du problème.

Q3. Un code rigoureux en début d'énoncé est attendu. Il fallait pour cette question attacher un grand soin à la terminaison de la fonction. En particulier, le dépassement de la borne supérieure de tableau, en testant la condition $s + l(m) \leq l(t)$ (précisée dans la question 1), doit être contrôlé.

Q4. Un exemple de complexité dans le pire cas (et dans le meilleur cas) doit être proposé dans le cas où $l(t)$ et $l(m)$ sont quelconques.

Un exemple correspondant à toutes les valeurs possibles des longueurs et non pour quelques valeurs.

Q5. Une complexité linéaire est imposée par l'énoncé. Le code donné doit donc en tenir compte, cela n'a pas été le cas dans un grand nombre de copies.

Q7. L'exemple demandé doit être détaillé et non survolé. En particulier, la condition de terminaison du programme doit être mise en évidence.

Q8. Un manque de rigueur dans les réponses à cette question a été constaté. En particulier, les deux cas doivent être traités avec soin.

Q10. Les mêmes problèmes que ceux identifiés pour la question Q4 sont apparus pour cette question pour le choix de l'exemple donné.

Q11 à Q15. Les valeurs de complexité qui sont demandées sont parfois fausses ou non données. Beaucoup de candidats ne considèrent pas qu'une complexité qui dépend seulement de la taille de l'alphabet peut être vue comme constante. Pourtant, il est bien précisé dans l'énoncé que cette taille est égale à 26.

Q21. Beaucoup trop d'erreurs. Très souvent, la détermination d'un automate ne semble pas maîtrisée.

Q22. La preuve de validité est rarement faite. De plus, l'algorithme donné n'est pas de complexité $O(l(t))$ imposée par l'énoncé.