



## 1/ Remarques générales

Le sujet comporte deux parties indépendantes.

### 1. Erreurs courantes

Confusion entre type et expression en Caml.

### 2. Remarques sur le texte, sa compréhension

Les ambiguïtés sur la définition 4 de « formule » dans la partie 1 ont entraîné des soucis pour certains candidats.

Il aurait été intéressant de donner plus d'exemples dans la partie 2.

### 3. Réactions (prévues ou non)

Choix de la récursivité en question II.11 qui a pu déstabiliser certains candidats, suivant la structure de données choisie.

Parfois des réponses courtes, sans justification.

### 4. Bilan

La longueur et la difficulté du sujet ont semblé parfaitement adaptées aux candidats. La correction s'est avérée plutôt aisée.

## 2/ Remarques détaillées par question

### PARTIE I

Partie classique, assez facile (surtout questions 1 à 7, puis 10 et 11).

Beaucoup d'oublis du mot « Var » dans la réponse à la question 1.10 et des constantes dans la question 1.11.

**Question 1.1** : facile et bien traitée.

**Question 1.2** : facile et bien traitée. Un certain nombre de candidats oublie toutefois de justifier leur réponse. Quelques réponses inutilement alambiquées. Au contraire, très peu de candidats ont osé fournir la réponse simple T.

**Question 1.3** : moyennement bien traitée. De nombreux candidats ont donné les réponses simples min et max. Quelques autres réponses plus originales.

**Questions 1.4 à 1.7** : faciles et bien traitées.

**Question 1.8** : peu traitée. Un très faible nombre de candidats a donné l'induction structurelle sur les formules attendues (un meilleur taux de succès aurait peut-être été obtenu si la structure en question avait été préalablement définie dans l'énoncé).

**Question 1.9** : facile mais dont la formulation semble avoir dérouté la majorité des candidats. Très peu de réponses correctes.

**Question 1.10** : facile et bien traitée par les candidats ayant une maîtrise raisonnable du langage Caml. Beaucoup d'oublis du Var.

**Question 1.11** : un certain nombre de candidats confondent « renvoyer une chaîne de caractère » avec « renvoyer void et a cher du texte ». L'opérateur ^ pour la concaténation de chaînes de caractères ne semble pas non plus connu de tous les candidats. On notera surtout que la quasi-totalité des candidats n'a pas jugé utile de parenthéser les chaînes de caractères renvoyées.

## PARTIE II

### Remarques générales

- Beaucoup trop d'utilisations de ref et de for.
- Difficulté pour certains à définir un réseau de transport.
- Question 11, il est demandé une implémentation récursive. Suivant le type de données choisi en question 3, il peut être incongru d'utiliser la récursivité.
- Très peu de réponses sur la complexité.
- Question 2.5, il est suggéré l'utilisation d'une pile. Certains l'ont redéfini, d'autres ont utilisé un type liste, d'autres ont traité la question autrement. La notation a été adaptée.
- Dernière question très rarement traitée.

### Remarques par questions

**Question 2.1** : facile. Quelques candidats se contentent de paraphraser.

**Question 2.2** : délicate, traitée par la majorité des candidats qui fournissent le plus souvent une preuve outrageusement fausse.

**Question 2.3** : relativement bien traitée. De nombreux candidats semblent cependant ne pas bien maîtriser la syntaxe des définitions de types dans le langage Caml. Plusieurs choix possibles pour la représentation des structures de données. La question comportait de nombreuses sous-questions, certains éléments ont parfois été oubliés. Deux solutions cohabitent pour la représentation des arcs infinis : l'usage d'une valeur négative (imposant le traitement d'un cas particulier dans les programmes des questions suivantes, parfois négligé) et l'usage d'une grande valeur.

**Question 2.4** : écriture d'un programme assez long mais relativement bien traitée.

**Question 2.5** : difficile et souvent ignorée. De nombreux programmes plus ou moins incorrects et quelques bonnes réponses.

**Question 2.6** : facile. Certains candidats oublient toutefois quelques éléments de réponse. Les réponses données sont souvent plus verbeuses que nécessaires.

**Question 2.7** : relativement bien traitée.

**Question 2.8** : l'algorithme donné pouvant ne pas terminer [1], la question a été (partiellement) banalisée.

**Question 2.9** : extrêmement peu de réponses correctes.

**Question 2.10** : facile et bien traitée.

**Question 2.11** : plus ou moins traitée. L'essentiel des réponses néglige la consigne imposant un style de programmation fonctionnelle. Suivant le type de données choisi en question 3, il a semblé incongru à certains candidats d'utiliser la récursivité.

**Question 2.12** : facile, bien traitée par les candidats l'ayant remarquée.

**Question 2.13** : de nombreuses bonnes réponses de la part des candidats l'ayant abordée.

**Questions 2.14 à 2.18** : questions moins traitées.

Quelques bonnes réponses aux questions 2.15 et 2.17, relativement simples.

**Question 19** : très rarement traitée.

[1] Uri Zwick, The smallest networks on which the Ford-Fulkerson maximum flow procedure may fail to terminate, Theoretical Computer Science, 1995, [http://dx.doi.org/10.1016/0304-3975\(95\)00022-O](http://dx.doi.org/10.1016/0304-3975(95)00022-O)

## Conclusion

Première partie très classique, sans grande difficulté.

Seconde partie très orientée graphe et algorithmes avec des possibilités pour les candidats de s'exprimer sur la partie algorithmique, codage et modélisation.

La longueur du sujet paraît raisonnable pour l'ensemble des candidats. Il semble que l'épreuve n'ait pas été trop difficile et ait permis de repérer les candidats faibles en informatique.