

## 1/ PRESENTATION DE L'ÉPREUVE

L'épreuve portait sur l'analyse et le traitement d'une trame GPS provenant d'un satellite via un récepteur pouvant équiper un véhicule dans le but de l'afficher sur une carte. Le sujet était décomposé en 5 parties :

- la première partie du sujet portait sur la lecture de cette trame sur le port série puis de son enregistrement dans un fichier ;
- la seconde partie portait sur l'analyse du fichier et la réorganisation des informations de la trame dans une liste de points ;
- la troisième partie portait sur la réorganisation des points dans une structure évoluée ;
- la quatrième partie portait sur l'affichage du parcours GPS sur une carte ;
- la cinquième partie demandait aux candidats de réaliser une synthèse du travail effectué.

Les bases de données n'étaient pas abordées dans cette épreuve. Le calcul scientifique l'était par le biais du calcul de moyennes glissantes et du calcul de la couleur du point.

## 2/ REMARQUES GÉNÉRALES

Pour cette première épreuve d'informatique, la volonté était de proposer une épreuve très largement accessible aux candidats de la filière TSI. Les correcteurs pouvaient constater un niveau très hétérogène des candidats : pour certains, l'épreuve n'était pas ou peu traitée ; pour d'autres l'épreuve était entièrement maîtrisée. Ainsi, l'étalement des notes de cette épreuve permettait de très bien classer les candidats.

La volonté de cette épreuve était de se baser sur la résolution d'un problème informatique pouvant s'appliquer à n'importe quelle discipline scientifique. Au cours de l'épreuve, le candidat devait donc répondre à des objectifs intermédiaires dont le but final était de répondre au problème initial.

Malgré une certaine bienveillance lors de la correction des copies, deux problèmes majeurs étaient constatés. D'une part, on remarquait une mauvaise maîtrise des structures algorithmiques élémentaires et des confusions dans l'utilisation des fonctions (mélange entre définition d'une fonction, appel d'une fonction, variable locale peu ou pas déclarée...). D'autre part, la majorité des candidats était dans l'incapacité de prendre du recul sur le problème posé. Les réponses attendues étaient souvent fournies dans le texte ou dans les annexes, mais une grande partie des candidats était passé à côté.

Enfin, sur un peu plus de 900 copies, moins de 20 candidats composaient en Scilab.

## 3/ REMARQUES SPÉCIFIQUES

### Question 1.

Cette question ne présentait pas de difficulté particulière.

Les caractères étant codés en ASCII, il suffisait de compter les octets utiles dans la trame donnée en annexe.

### Question 2.

Cette question ne présentait pas de difficulté particulière.

Beaucoup de candidats oubliaient qu'il était précisé dans le texte que 10 bits étaient nécessaires à la transmission d'un octet dans le cadre d'une transmission GPRMC. Beaucoup se trompaient dans le calcul du rapport permettant d'évaluer la durée de transmission.

**Question 3.**

Cette question ne présentait pas de difficulté particulière. Il suffisait d'utiliser la fonction `initSerie()` en utilisant les paramètres cités dans le texte.

**Question 4.**

Cette question présentait peu de difficulté.

Pour donner le code de <LF> certains candidats donnaient successivement les codes de <, L, F et >.

**Question 5.**

Cette question semblait ne pas présenter de difficulté. Cependant, beaucoup de candidats n'étaient pas en mesure de traduire directement un algorithme en pseudo code en langage Python ou Scilab. Les efforts pour marquer les indentations n'étaient pas toujours faits. Alors qu'il était explicitement demandé de documenter la fonction, beaucoup de candidats omettaient de le faire.

**Question 6.**

Il s'agissait de faire une « mini synthèse » du travail effectué en réutilisant les fonctions définies précédemment. Peu de candidats avaient le recul nécessaire pour utiliser les fonctions des précédentes questions. Il était décidé par les auteurs de mettre les documentations complètes des fonctions. Beaucoup de candidats recopiaient tous les arguments de la méthode `open()` sans réfléchir à ceux qui étaient réellement nécessaires.

**Question 7.**

Il s'agissait dans cette question de vérifier que les candidats étaient en mesure de valider le fonctionnement d'un algorithme simple. Une majorité de candidats ne détectait pas l'absence de l'incrémentement du `i`. Beaucoup pensaient que `else` était oublié alors qu'il n'était pas nécessaire.

Une grande proportion de candidats ne faisaient pas attention à l'indentation, voire des confusions dans le vocabulaire utilisé : incrémentement, indentation, itération etc.

**Question 8.**

Cette question nécessitait de compléter une fonction très proche de celle qui était proposée précédemment. Beaucoup de candidats faisaient la confusion entre la valeur retournée et l'indice de la valeur.

Dans le cas où le caractère n'était pas trouvé, beaucoup de candidats retournaient du texte, ce qui n'était pas le plus astucieux.

**Question 9.**

La première partie de la question ne posait globalement pas de problèmes.

Peu de candidats proposaient un algorithme (ou une fonction) pour répondre à la seconde partie de la question.

**Question 10.**

Tout d'abord, cette question demandait de compter la position des caractères. Une proportion non négligeable de candidats ne répondait pas à la question ou proposait des réponses complètement fantaisistes. Une très grande majorité des candidats oubliait de compter les signes de ponctuation dans leur décompte.

Enfin, une grande proportion des candidats n'était pas en mesure d'utiliser les fonctions proposées dans le sujet.

**Question 11.**

La question était peu abordée. Peu de candidats étaient en mesure de faire une synthèse sur les activités proposées au cours de cette partie.

**Questions 12, 13 et 14.**

Ces questions contenaient le seul pic de difficulté de ce sujet, mais étaient tout de même abordables.

Les candidats pouvaient être déstabilisés par la longueur de la documentation permettant de répondre aux questions.

Les quelques réponses présentes étaient en revanche parfaitement justes, ce qui montrait bien l'hétérogénéité des candidats.

**Question 15.**

A priori, cette question présentait peu de difficultés, mais peu de candidats étaient en mesure de donner de réponse juste.

**Question 16.**

Il s'agissait d'abord, d'étendre les résultats de la question précédente. Peu de candidats étaient en mesure de répondre.

Le calcul de la moyenne était peu traité. La formulation du document réponse déstabilisait peut-être les candidats : peu prenait l'initiative d'indenter le calcul de la somme puis de supprimer l'indentation pour le calcul de la division. Beaucoup de candidats réalisaient la somme des 4 termes sans tenir compte de la boucle.

**Question 17.**

La réponse attendue était évidente, mais peu de candidats y répondaient.

**Question 18.**

Question peu ou mal traitée.

**Question 19.**

À partir d'un cahier des charges précis, les candidats n'étaient pas en mesure d'implémenter la fonction demandée.

**Question 20.**

Pour cette question, il suffisait de remplir le programme principal en utilisant la documentation. Encore une fois, peu de candidats étaient en mesure d'appeler des fonctions données.

**Question 21.**

L'objectif de cette question était de réaliser une synthèse des activités proposées. Peu de candidats avaient le recul nécessaire pour aborder cette question.

**4/ CONSEILS AUX CANDIDATS**

Les correcteurs conseillent aux candidats des années futures d'avoir une meilleure maîtrise des concepts de base de la programmation (structures algorithmiques, définitions de fonctions, appels de fonctions...) et les encouragent à prendre davantage de recul pour traiter les épreuves proposées.