



1/ Présentation de l'épreuve

L'épreuve portait sur l'étude de certaines fonctionnalités d'une application sportive « Rugby Manager ».

Le sujet était décomposé en 4 parties totalement indépendantes :

- la première partie du sujet portait sur le traitement de l'image et sur le traitement de statistiques ;
- la seconde partie portait sur l'utilisation de bases de données ;
- la troisième partie portait sur le tri de listes issues de ces bases de données et sur l'espace mémoire utilisé ;
- la quatrième et dernière partie abordait des notions de simulation numérique élémentaires avec la modélisation de la passe vissée.

2/ Remarques générales

Les correcteurs constatent que les bases de l'algorithmique ne sont toujours pas maîtrisées par la majorité des candidats. La manipulation des tableaux ou des listes est beaucoup trop floue pour une trop grande partie d'entre eux.

Les notions de variables, caractères, entrées/sorties de fonctions, types, ne sont toujours pas maîtrisées. Notons que le type « *Bouhilien* » n'existe pas, surtout quand il s'agit d'une liste.

Certes, la difficulté du sujet était irrégulière mais il est regrettable que beaucoup de candidats n'aient pas su répondre aux questions très faciles dispersées dans le sujet. Les différentes parties d'un sujet étant indépendantes, les candidats ont intérêt à s'intéresser à chacune d'entre elles afin de mettre en évidence l'étendue de leurs compétences. Cette année, les quelques questions très faciles permettant de vérifier les compétences « savoir lire », « savoir s'approprier le contenu d'un sujet » ont permis de constater qu'elles ne sont pas encore acquises par tous les candidats, malgré leur caractère trivial.

On note cependant une progression dans l'utilisation des fonctions et des variables pour une bonne moitié de candidats.

On constate également une progression régulière sur les connaissances en Bases de Données, qui ont été plutôt bien abordées par la majorité des candidats.

2.1/ Remarques générales sur le sujet

Le sujet proposé valide les compétences du programme :

- analyser et modéliser un problème, une situation ;
- imaginer et concevoir ;
- spécifier rigoureusement les modules ou fonctions ;
- évaluer, contrôler, valider des algorithmes et des programmes ;
- communiquer à l'écrit une solution ou un algorithme.

Partie 1

Les questions de cette partie sur le traitement d'images utilisées dans l'application étaient de difficultés très variables. Outre les questions très simples pour lesquelles il suffisait de comprendre l'annexe, cette partie

permettait de mettre en évidence les compétences des candidats en algorithmique. Certaines questions se sont révélées très sélectives (Q6, Q7).

→Référence au programme :

- algorithmique et programmation I
- algorithmique et programmation II

Partie 2

Partie traitant sur les bases de données utilisées dans l'application. L'objectif n'était pas de porter une réflexion sur la structure de la base, mais plutôt de s'intéresser aux requêtes.

→Référence au programme : initiation aux bases de données

Partie 3

Cette partie proposait de trier des listes issues des bases de données et de travailler sur l'espace mémoire utilisé. Les algorithmes de tri étaient donnés et les questions visaient la compréhension de ces fonctions, voire leur modification. Ces questions ne présentaient pas de réelle difficulté mais assez peu de candidats ont réussi à bien les traiter. Le codage des nombres entiers et flottants était également abordé dans cette partie : il apparaît que cette normalisation semble très floue pour un grand nombre de candidats.

→Référence au programme :

- algorithmique et programmation II
- représentation des nombres et conséquences

Partie 4

Partie relativement simple sur une équation différentielle du premier ordre modélisant la vitesse du ballon de rugby. Les compétences sur la création de listes simples et sur la méthode d'Euler étaient testées et on s'aperçoit que beaucoup de candidats ont du mal à bien structurer leur raisonnement. Les dernières questions de cette partie abordaient les fonctions d'affichage, qui sont indispensables en informatique.

→Référence au programme : ingénierie numérique et simulation

2.2/ Remarques générales sur la composition des candidats

En ce qui concerne l'écriture des algorithmes, on peut noter un manque de rigueur dans un grand nombre de copies. Lorsque que l'algorithme est long ou possède de nombreuses variables, il est judicieux d'utiliser des commentaires afin de faciliter la compréhension.

Le passage des paramètres n'est toujours pas compris par une grande majorité, on constate beaucoup de confusions sur les types des variables utilisées (dans la définition des boucles *for* par exemple).

La majorité des candidats est dans l'incapacité de prendre du recul sur le problème posé. Des éléments importants dans les réponses attendues sont souvent dans le texte ou dans les annexes, mais une grande partie des candidats passe à côté de ces informations.

Cependant, nous avons été heureux de voir quelques candidats remarquables qui ont su montrer leur aisance en informatique.

Enfin, une trop grande majorité des candidats perd encore des points sur la présentation avec un usage abusif du correcteur liquide ou ruban qui nuit à la lisibilité des algorithmes.

3/ Remarques spécifiques

- Q1. Question très simple réussie par la majorité des candidats. On constate toujours qu'il reste quand même quelques candidats qui ne comprennent pas l'énoncé ou recopient l'annexe sans l'adapter. Il fallait travailler dans le bon répertoire de travail pour ouvrir l'image.

- Q2. Là encore, question très simple réussie par la majorité des candidats, bien que certains aient recopié l'annexe sans l'adapter. Le travail sur les dimensions d'un tableau reste encore un peu difficile pour certains et il faut faire attention aux types de variables utilisées dans la fonction `print`.
- Q3. Il s'agissait ici de vérifier que les candidats étaient capables de récupérer une valeur dans un tableau à partir d'indices calculés. Beaucoup oublient que ces indices doivent être entiers et les bases syntaxiques des fonctions sont encore mal maîtrisées pour un grand nombre de candidats.
- Q4. On constate un manque de connaissances en architecture et en implémentation des nombres entiers.
- Q5. Beaucoup de candidats n'ont pas compris la question : l'objectif était de réutiliser la fonction précédente et le tableau image.
- Q6. Question difficile traitée par peu de candidats. Les propositions de certains candidats étaient cependant intéressantes.
- Q7. Question difficile. Il suffisait de reprendre la question 6 et l'adapter.
- Q8. Seulement 1/3 des candidats ont traité cette fonction qui n'était pourtant pas d'une grande difficulté.
- Q9. Il fallait réutiliser les fonctions définies précédemment. Notons que certains candidats n'ayant pas traité les questions précédentes ont quand même réussi à répondre à cette question.
- Q10. Problème de compréhension, là encore on peut noter une insuffisance sur la manipulation des listes et tableaux. Les notions de *slicing* ne sont manifestement pas acquises.
- Q11. Question difficile, ceux qui ont compris la question ont proposé des solutions intéressantes, voire originales.
- Q12. Cette question relativement simple mais essentielle abordait des notions classiques d'algorithmique qui devraient être connues de tous les candidats. Il est très surprenant de constater que tous les candidats ne sont pas capables de trouver un minimum, un maximum et une moyenne à partir d'une liste d'entiers. De plus, il est anormal d'utiliser 3 boucles *for* pour parcourir 3 fois la même liste afin de déterminer ces trois valeurs. Ce choix n'a pas été sanctionné cette année. Pour l'avenir, rappelons que le milieu d'un intervalle n'est pas la moyenne ! ... « *le monde n'est pas linéaire !* »
- Q13. Requête très simple sur les bases de données, bien traitée dans l'ensemble. Certaines réponses ont été données en algèbre relationnel, ce qui n'était pas demandé.
- Q14. Mêmes remarques que la question précédente.
- Q15. Requête plus complexe, avec une sélection en amont d'une fonction d'agrégat, peu de candidats ont donné une réponse complète.
- Q16. Il suffisait de comprendre la sortie d'un programme et le type liste qui allait être trié par la suite. En prenant du recul sur les questions suivantes, qui triaient la liste, la réponse était évidente.
- Q17. Il fallait corriger deux lignes d'un algorithme classique de tri (*tri par sélection*). On constate qu'une moitié des candidats ne maîtrise pas ou ne comprend pas un algorithme aussi simple.
- Q18. Question facile sur la complexité de 2 boucles imbriquées. De nombreux candidats ont donné la bonne réponse, sans forcément bien la justifier.

- Q19. Commentaires demandés sur un algorithme proposé. On constate que l'analyse d'algorithmes qui ont dû être abordés en cours ne sont pas connus et compris. Beaucoup de candidats se contentent d'expliquer chaque mot du code sans réellement prendre du recul en cherchant à montrer l'utilité de la séquence. Il fallait notamment souligner la récursivité de la fonction, le test d'arrêt et la séparation de la liste à l'aide du pivot. La deuxième partie de la question était plus complexe, il fallait trouver une solution pour compter les appels successifs de fonction. Certains candidats ont vu le compteur mais pas le côté global de cette variable. La solution envisagée était de mettre en sortie de fonction ce nombre de coups mais quasiment aucun candidat n'y a pensé.
- Q20. On constate encore dans cette question que trop de candidats ne comprennent pas les variables d'entrée d'une fonction.
- Q21. Question traitée par pratiquement tous les candidats avec un bon taux de réussite pour la partie entière. Le codage de la partie décimale est au contraire méconnu de la plupart des candidats.
- Q22. Bien qu'il semble logique que le codage au format IEEE 754 soit abordé en classe et malgré les rappels, très peu de candidats ont su coder correctement l'exposant biaisé et la mantisse.
- Q23. Le calcul de l'espace mémoire a été bien traité par une grande partie des candidats, quelques erreurs de calcul dommageables cependant.
- Q24. Pas de difficulté dans cette question si ce n'est que beaucoup de candidats utilisent une boucle *for* avec un *range* sans se préoccuper du pas qui doit être entier, c'est une grosse erreur.
- Q25. Question de cours sur la réponse analytique à une équation différentielle du premier ordre, qui a posé beaucoup plus de problème que prévu ! La fonction Python qui en découle est triviale.
- Q26. Ici la méthode d'Euler n'était pas explicitée mais une connaissance parfaite de cette méthode par approximation était exigée. Trop peu de candidats arrivent à mettre correctement en place cette méthode pourtant classique.
- Q27. Il s'agissait dans cette question de réutiliser les 2 fonctions précédentes à l'aide d'une simple boucle. Les candidats qui l'ont traitée ont réussi à bien comprendre l'objectif.
- Q28. La dernière question faisait référence au module graphique de Python afin de tracer les sorties des fonctions définies précédemment, ce qui est classique.

Précisons enfin que, comme chaque année, quelques points du barème sont attribués pour le soin apporté aux copies, pour l'orthographe et pour l'écriture des algorithmes et programmes en langage Python.

4/ Conseils aux candidats

Les correcteurs continuent de conseiller aux candidats d'avoir une meilleure maîtrise des concepts de base de la programmation (structures algorithmiques, définitions de fonctions, appels de fonctions, fonctions récursives, ...), une meilleure maîtrise du langage de programmation Python (syntaxes de base).

Les correcteurs encouragent les candidats à prendre davantage de recul pour traiter les épreuves proposées : ne pas bloquer à la première difficulté et anticiper la suite de l'énoncé afin de répondre à certaines questions.

