PSI

CONCOURS COMMUN INP 2025

RAPPORT DE L'ÉPREUVE ÉCRITE D'INFORMATIQUE

1/ COMMENTAIRE GENERAL SUR L'ÉPREUVE

L'épreuve d'informatique du concours CCINP proposait de travailler sur la gestion de randonnées. La première partie traitait de la gestion des randonnées à l'aide d'une base de données. La deuxième partie traitait du calcul du dénivelé des randonnées. La troisième partie traitait de l'organisation d'un trek pour rechercher le chemin le plus court en termes de difficultés.

Le sujet permettait de balayer un large ensemble des compétences décrites dans le nouveau programme d'informatique en CPGE.

Cette session a poursuivi la mise en place d'un document réponse. Les candidats l'ont globalement bien utilisé. Il est rappelé qu'il est utile de réfléchir sur un brouillon afin de porter le plus proprement possible la solution sur le document réponse. La qualité de la présentation et de la rédaction est prise en compte dans la notation.

Dans l'ensemble, les questions de SQL ont été bien traitées, bien qu'on observe assez fréquemment des imprécisions sur les mots clés (`WHILE` au lieu de `WHERE`, `ORDRE BY` au lieu de `ORDER BY`) ainsi que l'ordre de référence d'un attribut dans une table (`Attribut.Table` plutôt que `Table.Attribut`). Les candidats se sont dans l'ensemble investis dans l'apprentissage du SQL et de nombreux points y ont été gagnés. On note également beaucoup de confusions entre `HAVING` et `WHERE`.

À l'inverse, les candidats qui n'ont pas fait l'effort de s'y plonger se trouvent très vite démasqués et cela n'augure en général rien de bon pour la suite de l'épreuve.

Pour la lisibilité des programmes, il est vivement conseillé de choisir des noms de variables intelligibles, de bien soigner son indentation avec l'alignement sur les carreaux ou un trait vertical. En revanche, on évitera absolument d'utiliser un caractère censé représenter l'indentation et répété sur toute la ligne, que ce soit un trait horizontal, un point ou le caractère ␣. La lecture d'un tel code par le correcteur est très difficile et ne peut que désavantager le candidat.

Pour les questions en SQL, il pourrait être intéressant d'écrire une ligne par mot clé pour assurer la lisibilité de la requête.

L'utilisation des commentaires, bien qu'appréciée, ne doit pas surcharger la copie et faire perdre trop de temps aux candidats.

Il était écrit, avec un exemple, de ne pas recopier les signatures des fonctions qui sont présentes pour aider le candidat à comprendre le type des données d'entrées et sorties. Pourtant, un certain nombre de candidats les ont recopiées et même certains ont utilisé des syntaxes comme `altitude_maximale(iti :itineraire) -> float – itinéraire[0][2].`

Cette session semble marquer une progression quant à la maîtrise de Python par rapport à la session précédente.

Les correcteurs rappellent aux candidats que seule une pratique régulière du langage sur machine peut leur permettre de mettre en place un certain nombre d'automatismes dont l'absence est flagrante dans de nombreuses copies. Il est utile d'essayer de se réentraîner sur quelques anciens TP sur machine lors de la préparation aux écrits pour que ces automatismes reviennent juste avant les épreuves. En outre, s'obliger à écrire systématiquement le code sur feuille pendant toute l'année avant de le taper à l'ordinateur pendant les TP vous permettra aussi de soigner la présentation et la lisibilité de celui-ci.

De manière non exhaustive, voici quelques erreurs relevées au gré des copies :

- `range 2` ou `range([0 :10])`
- `a = b` au lieu de `b = a`
- `If, While, Else`, etc. avec une majuscule
- Non parenthésage dans les divisions `somme / j+1` est différent de `somme / (j+1)`
- Condition d'égalité avec un seul symbole `=` ou affectation avec `==`
- Structure : `valeur if condition` avec un `else` sans instruction ou des instructions inutiles `max = max`…
- Utilisation de nom de variable non valide : alt_i, alt_{i+1}, `alt'`

Pour les questions d'algorithmique simple, il n'est pas attendu que le candidat utilise les fonctions `max, sort` surtout dans les premières questions qui visent à tester les compétences de bases en Python.

2/ ANALYSE DÉTAILLÉE DES QUESTIONS

Q1 : Le fait que `Titre` soit une chaîne de caractères n'est pas suffisant pour justifier que cela ne peut pas être une clé primaire.

Q2 : Quand la notion de clé étrangère est connue, cette question ne pose pas de difficulté.

Q3 : La sélection des randonnées « à pied » a régulièrement été oublié. De manière générale, l'ordre des mots clés à une importance dans la requête.

Q4 : Le `GROUP BY` a été régulièrement oublié pour regrouper les résultats associés à chaque auteur.

Q5 : La jointure est assez bien réalisée mais attention a bien repérer l'identifiant de l'auteur avec `Auteur.Id` pour ne pas avoir de confusion avec `Randonnee.Id`

Q6 : Cette question, plus difficile, est plutôt mal traitée avec la combinaison de beaucoup de mots clés.

Q7 : Quelques candidats n'arrivent pas à réaliser un import de module propre, un simple `import gpxpy` était attendu.

Q8 : Cette question est en générale correctement traitée.

Q9 : La réponse attendue doit faire apparaître « complexité linéaire » ou O(n).

Q10 : Cette première question de code, réalisée parfaitement par la moitié des candidats, demande une simple détermination de maximum : il faut faire attention à l'initialisation, notamment il est possible de réaliser des randonnées dans des zones à altitude négative… On note quelques problèmes d'extraction de la valeur de l'altitude dans la liste de tuple.

Q11 : Quelques candidats ont du mal à utiliser un appel correct à la fonction précédente.

Q12 : Cette question est plutôt bien traitée malgré quelques erreurs sur les indices du range.

Q13 : Cette question un peu plus difficile a été moyennement bien traitée. Les candidats se sont mélangés entre les notations de l'énoncé et les variables qu'ils ont définies dans leur fonction et ont mis des bornes fausses pour le calcul de la moyenne...

Q14 : Un nombre important de candidats a proposé une complexité faisant apparaître n et p, outre la bonne réponse, nous avons vu à peu près toutes les possibilités : n^p, n log(p), n+p, n/p, 1/n… Il vaut parfois mieux ne rien écrire, surtout quand on n'a pas traité la question précédente !

Q15 : Il fallait bien préciser qu'il s'agissait des listes de latitude et longitude contenues dans le dictionnaire `dem`.

Q16 : Beaucoup de candidats ne comprennent pas ce qu'est la signature d'une fonction, pourtant rappelée au début du sujet qu'il est conseillé de lire attentivement. Les candidats qui ont tenté la question l'ont globalement réussie. On rappelle qu'il est attendu de donner le type des arguments d'entrée et le type de la sortie.

Q17 : Question assez bien traitée. Des points négatifs sont prévus pour les cochages aléatoires pouvant ramener la note à 0 point pour cette question.

Q18 : Il y a un cas de base ou « un `return` donc ça se termine » n'est pas une justification correcte. Un mot disant que la fonction auxiliaire se termine également était également attendu.

Q19 : Le nom de tri fusion ou partition fusion n'est pas reconnu par les candidats, seule la moitié d'entre eux ont les points. Ce tri classique faisant partie des tris généralement étudiés en 1re année, les correcteurs auraient espéré une meilleure réussite.

Q20 : Fonction assez correctement traitée par les candidats ayant compris. En revanche, beaucoup de candidats calculent l'indice « milieu » en faisant une différence entre `ind_fin` et `ind_deb`… On note également un grand nombre de divisions non entières alors que le résultat doit être un entier.

Q21 : Question peu traitée mais assez correctement quand elle l'a été. De trop nombreux candidats modifient cependant la liste passée en argument alors qu'il était demandé un nouvel itinéraire. On retrouve également beaucoup de L.append(a,b,c) au lieu de L.append((a,b,c)).

Q22 : Il n'était pas demandé de décrire le programme ligne à ligne mais d'expliquer ce qu'il faisait sur les données en une phrase simple. Beaucoup de candidats reconnaissent un algorithme glouton.

Q23 : Trop de candidats tentent de répondre en lisant le graphe plutôt qu'en essayant de comprendre ce que fait l'algorithme. C'est dommage car finalement toute la suite découle de cette compréhension.

Q24 : Un contre exemple est le meilleur moyen de montrer le problème de l'algorithme glouton.

Q25 : Cette question est assez bien traitée malgré quelques erreurs.

Q26 : Question bien traitée par les candidats ayant compris la fonction.

Q27 : Assez peu de bonnes réponses ici avec très peu de candidats qui la traitent, quelques mots clés ont été donnés : mémoisation, dictionnaire sans forcément expliquer.

Q28 : Question assez bien traitée.

Q29 : Question globalement bien traitée avec quelques erreurs dans quelques cases.

Q30 : Question traitée aléatoirement alors qu'il suffisait de prendre un contre exemple dans le tableau précédent.

Q31 : Question moyennement bien traitée, notamment les trois dernières lignes où ce sont souvent de mauvais noms de variables qui apparaissent.

Q32 à **Q34** : Questions théoriques assez peu traitées par les candidats, le plus souvent par les meilleurs, qui ont eu le temps d'arriver jusque là.

La correction partielle d'un programme ne consiste pas à corriger un programme faux !