



CONCOURS COMMUN INP RAPPORT DE L'ÉPREUVE ÉCRITE INFORMATIQUE Session 2026

1/ PRÉSENTATION DU SUJET

Autour de jeux

Le sujet portait sur le problème classique des n reines (ou des n dames). Il permettait d'explorer une grande partie du programme d'informatique de la filière TSI.

Dans l'ensemble, il a été bien compris, certains candidats faisant même preuve de solides compétences en le traitant presque intégralement. Les résultats sont donc très positifs et encourageants.

Il subsiste cependant un certain nombre de candidats qui n'ont pas consacré suffisamment de temps à travailler cette discipline durant leurs années de classe préparatoire. Ceux-ci n'ont pas pu donner le change sur ce sujet.

2/ REMARQUES GÉNÉRALES

Erreurs les plus fréquentes

- dictionnaires : la structure de dictionnaire en Python reste assez mal comprise par certains candidats, tout particulièrement l'insertion d'un couple (clé, valeur) ;
- modules : l'importation et l'exploitation des modules ne sont pas maîtrisées par un nombre important de candidats ;
- listes : les manipulations de base sur les listes restent encore une source d'erreurs chez certains ;
- boucles : la gestion des bornes dans les boucles *for* est encore parfois approximative ;
- listes en compréhension : ce concept syntaxique n'est pas encore pleinement assimilé par une partie des candidats. Il pourrait être davantage utilisé (sauf lorsqu'il est explicitement interdit comme en question **Q19**) ;
- erreurs de logique : par exemple, la mauvaise position des *return* (indentation).

Remarques sur le sujet, le texte et sa compréhension

Il n'y a pas eu de difficultés de compréhension du sujet et pas de remarques particulières.

Problèmes ou difficultés rencontrés lors de la correction

Bien que la numérisation améliore considérablement la lisibilité des copies (notamment grâce à la possibilité de zoomer), le candidat doit toujours soigner sa présentation. Certaines écritures restent illisibles ou trop petites, ce qui peut entraîner des confusions entre des caractères comme `i` ou `1`, `i` ou `l`, `1` ou `l`, etc.). Dans ces cas, la copie est pénalisée.

3/ REMARQUES SPÉCIFIQUES

Partie I - Des algorithmes pour résoudre le problème des n reines

- Q1.** Question d'appropriation du sujet. Globalement bien traitée.
- Q2.** Question d'appropriation du sujet. Bien traitée.
- Q3.** Globalement bien traitée. Cependant, cette question de base (créer une liste de n zéros) n'a pas été réussie par tous.
- Q4.** L'erreur la plus fréquente concernait la création de listes liées (`[ligne(n)]*n` ou `G.append(L)` en boucle sans créer une nouvelle liste à chaque tour). Il s'agit d'une finesse du langage Python qu'il faut maîtriser, les listes de listes étant un outil de base dans de nombreux problèmes.
- Q5.** La difficulté résidait dans l'évitement d'une double boucle pour avoir une complexité en $O(n)$.
- Q6.** Pourtant facile, cette question a perturbé un nombre non négligeable de candidats. L'annexe 2 du sujet contenait pourtant toutes les informations nécessaires sur la fonction `print`.
- Q7.** Question généralement bien traitée. Aucune justification approfondie n'était attendue (notamment sur le respect de (b) et (c)). On note parfois des confusions entre lignes et colonnes.
- Q8.** L'évaluation portait sur deux points : la maîtrise des dictionnaires et la bonne implémentation logique de la démarche. On constate fréquemment des erreurs lors de l'insertion/l'ajout d'un couple (`cle`, `valeur`) dans le dictionnaire. Rappelons que la méthode `append` est spécifique aux listes Python.
- Q9.** Il s'agissait de conserver la structure de la fonction tout en corrigeant deux erreurs : la gestion des indices et l'inversion de `True` et de `False`). Les deux corrections attendues étaient rarement présentes toutes les deux. L'erreur sur `range(i, n)` au lieu de `range(i+1, n)` était très fréquente. Certains candidats ont ajouté une erreur en modifiant l'indentation des `return`. Pour la complexité de `tous_distincts`, les correcteurs auraient apprécié que davantage de candidats soulignent que le test d'appartenance à un dictionnaire est en $O(1)$ (comme précisé en annexe 2).
- Q10.** Deux étapes étaient nécessaires pour bien répondre : construire les listes requises, puis effectuer les bons tests (utilisation de l'opérateur `and` ou bon enchaînement de `if...`). C'est la première étape qui a été le plus souvent mal traitée.
- Q11.** Cette question simple a généralement été bien traitée, sauf par les candidats ayant oublié le test sur la longueur de `L`.
- Q12.** Savoir importer correctement des modules Python est une compétence de base. Pourtant, cette question, qui aurait dû être très facile pour tous les candidats, a donné lieu à un nombre assez important de réponses partiellement ou totalement fausses.

Q13. Des erreurs de logique ont été observées chez certains candidats concernant la position des `return`. Une autre erreur classique observée consistait à générer une permutation avec `sample(L, len(L))`, à la tester, puis à renvoyer `sample(L, len(L))` en cas de succès (c'est-à-dire une nouvelle permutation non testée !).

Q14. Question classique mais assez mal traitée. Le *slicing* était autorisé mais peu utilisé. Parmi les candidats ayant construit une démarche cohérente, c'est souvent la gestion des indices qui a posé problème.

Q15. Il s'agissait d'exécuter pas à pas un code. Cela a pris un peu de temps et cela a été valorisé.

Q16. Question assez facile pour ceux qui avaient bien compris la question **Q15**. Peu de bonnes réponses ont été observées.

Q17. Question simple pour les candidats qui avaient bien compris la démarche de la sous-partie. L'oubli du paramètre dans l'appel à `est_solution` était l'erreur la plus récurrente.

Q18. Bien traitée dans l'ensemble.

Q19. Question classique (renverser/retourner une liste sans utiliser du *slicing*), pas si souvent bien traitée.

Q20. Question plus difficile, avec assez peu de bonnes réponses. Les candidats ayant bien répondu ont généralement commencé par créer une liste de n zéros avant de la remplir avec un parcours.

Q21. Question d'analyse de code. Le type de la variable renvoyée était « une liste ». Pour la structure de la variable, on attendait « liste de listes de configurations » ou « liste de listes de listes ». Enfin, expliquer le fonctionnement de la fonction n'a pas souvent donné lieu à des réponses claires. On attendait une ou deux phrases compréhensibles décrivant la démarche globale, et non un commentaire pas à pas de chaque instruction.

Q22. Il s'agissait de compléter une fonction récursive permettant un parcours du graphe des coups. En observant les codes des questions **Q22** et **Q23**, on pouvait déjà déduire une partie des réponses.

Q23. Même remarques que la question **Q22**.

Partie II - Base de données d'un serveur de jeu en ligne

Q24. Question simple sur les bases de données, souvent bien traitée.

Q25. Question avec une jointure, bien traitée dans l'ensemble.

Q26. Cette question était plus compliquée car elle nécessitait des jointures et des fonctions d'agrégations utilisées avec *GROUP BY*. Elle était logiquement discriminante.

Q27. Même remarque que la question **Q26**.

4/ CONCLUSION

Le sujet s'est avéré discriminant et a révélé un bon niveau général. De nombreuses questions étaient accessibles, avec un équilibre entre des questions particulièrement faciles et d'autres plus compliquées. La longueur du sujet a semblé adaptée et cohérente avec le temps imparti.