



TSI

CONCOURS COMMUN INP RAPPORT DE L'ÉPREUVE ÉCRITE INFORMATIQUE Session 2023

1/ PRÉSENTATION DU SUJET

Le sujet s'intéresse à l'analyse et à la programmation de méthodes de tests d'erreurs au sein d'une entreprise pour le numéro de sécurité sociale, le numéro de carte de crédit et les QR codes.

Il est composé de 22 questions réparties sur 5 parties :

- Partie I : Test du code de sécurité sociale.
- Partie II : Test de numéro de carte de crédit à l'aide de la formule de Luhn.
- Partie III : Test de QR Code.
- Partie IV : Gestion réseau. Partie sur le parcours du plus court chemin sur un graphe représentant les différents magasins de cette entreprise à l'aide de l'algorithme de Dijkstra.
- Partie V : Requêtes SQL. Partie permettant de tester différentes informations de cette entreprise.

2/ REMARQUES GÉNÉRALES

Dans la première partie, il s'agit de tester si un numéro de sécurité sociale est correct. Nous avons constaté que beaucoup de candidats ne savent pas encore manipuler correctement les listes. Le slicing par exemple, n'a été que trop peu compris malgré les annexes précisant leur fonctionnement.

Dans la seconde partie, on a pu constater que de nombreux candidats ne maîtrisaient pas la notion de dictionnaire.

La troisième partie a été très sélective, seuls les bons candidats ont réussi à traiter les différentes questions.

La quatrième partie, dédiée au parcours d'un graphe pour en déterminer le plus court chemin, a été difficilement traitée. Comme l'algorithme de Dijkstra n'est pas au programme de la filière TSI, l'énoncé présentait tous les points nécessaires à sa bonne compréhension.

La dernière partie, consacrée à la manipulation de requêtes SQL, a permis de constater de nombreux problèmes de syntaxe et de sémantique avec ce langage.

Le sujet fait très largement appel aux connaissances algorithmiques et pratiques du programme de première année. De nouvelles notions du nouveau programme de seconde année (dictionnaires, graphes, etc...) ont été également intégrées.

Erreurs les plus fréquentes des candidats

On constate une progression régulière sur les connaissances et sur les techniques de programmation du langage Python. Par exemple : la structure du code et les indentations sont bien respectées mais un certain nombre d'erreurs subsistent chez de nombreux candidats.

Les erreurs les plus fréquentes portent globalement sur :

- la manipulation des listes en général ;
- la manipulation du slicing mal comprise, surtout avec les indices négatifs ;
- la manipulation des chaînes de caractères, notamment des confusions entre le type *int* et le type *str* ;
- une mauvaise compréhension de la valeur retournée par une fonction ;
- la notion de copie superficielle ou profonde sur des listes de listes.

Bien que cela ne soit pas une erreur, trop de candidats utilisent la formulation suivante sur les manipulations des expressions logiques : *True if test==True else False* ou son équivalent sur 2 lignes.

Remarques sur le sujet, le texte et sa compréhension

Le sujet est de longueur et de difficulté adaptées, très peu de candidats sont parvenus à traiter quasiment l'intégralité du sujet.

La difficulté a été progressive dans chacune des cinq parties.

La troisième partie était la plus difficile, seuls les très bons candidats ont réussi à répondre correctement aux différentes questions.

Problèmes ou difficultés rencontrés lors de la correction

Parfois les noms de variables ne sont pas assez explicites et n'aident pas à la compréhension du code.

Le plus souvent on note une absence d'explication sur le code implémenté.

Les correcteurs ont fait preuve de bienveillance envers quelques erreurs de syntaxe, tant que le sens de la réponse restait compréhensible.

Un nombre non négligeable et croissant de candidats rendent des copies peu lisibles avec beaucoup de ratures.

Nous rappelons aux candidats que le soin apporté à la copie est évalué. Sont évalués :

- la concision, la clarté du code ;
- la présence de commentaires ;
- le style de programmation ;
- la qualité de la rédaction et de l'écriture.

Bilan

Les candidats ayant une bonne connaissance des techniques algorithmiques et une bonne maîtrise du langage Python trouvent le plus souvent des solutions simples aux questions posées. Ils arrivent à écrire en très peu de lignes de code Python le résultat demandé, ce qui n'est pas le cas des autres candidats.

3/ REMARQUES SPÉCIFIQUES

PARTIE I - TEST DU CODE DE SÉCURITÉ SOCIALE

Q1. Tous les candidats n'ont pas réussi à répondre à cette question particulièrement facile. Le principe de l'algorithme à implémenter en Python était pourtant totalement décrit dans la question.

Q2. En général bien réussie.

Q3. Des confusions sur les manipulations de chaînes de caractères et les nombres entiers chez certains candidats, sinon, question bien traitée.

Q4. Des erreurs ou omissions sur les conversions de type. Certains candidats n'ont pas acquis les opérateurs modulo et division entière.

PARTIE II - TEST DE NUMÉRO DE CARTE DE CRÉDIT

Q5. Beaucoup trop d'erreurs sur le type des variables à utiliser *str*, *int*.
Les itérations sur un entier sont souvent rencontrées.
Certains candidats oublient de renvoyer la liste dans le bon ordre.

Q6. Question difficile pour les candidats, de nombreuses erreurs de syntaxe. Beaucoup ont compris qu'il fallait tester la parité mais de fréquentes confusions élément/indice (test de parité effectué sur l'élément au lieu de l'indice).

Q7. Beaucoup trop de candidats ne connaissent pas la syntaxe des dictionnaires, notion pourtant bien définie dans le nouveau programme. L'accès à une valeur d'un dictionnaire est particulièrement mal maîtrisé.

Q8. Toujours des mélanges sur les types de variables (*str*, *int*), les candidats qui ont réalisé même imparfaitement cette question ont pu récupérer des points.

Q9. Question assez bien traitée dans l'ensemble.

PARTIE III - TEST DE QR CODE

Q10. Question pourtant facile mais, beaucoup trop de candidats n'arrivent pas à traiter correctement la structure d'une liste de listes.

Il y a également un grand nombre de candidats qui n'ont pas compris ce qu'est la copie de référence, en fait les notions de copies profondes et superficielles. Nous n'avons pas pénalisé ces candidats.

Les candidats qui ont utilisé la méthode `append` du module *List* pour la construction de leur liste en utilisant une double boucle ont souvent confondu `append(0)` avec `append([0])`.

Q11. Question assez bien réussie pour ceux qui l'ont traitée.
Des confusions chez certains entre le module et une fonction de ce module.

Q12. Question assez bien réussie pour ceux qui l'ont traitée.

Q13. Question difficile sur la manipulation de blocs de listes de listes. Question peu traitée avec les paramètres *x* et *y* souvent mal comprise.

Q14. Souvent bien traitée si abordée. Cependant des candidats se limitent à tester uniquement un seul bloc.

Q15. Question très sélective, ceux qui l'ont traitée ont assez bien réussi.

Q16. Question peu traitée, pas bien comprise avec beaucoup d'erreurs sur la taille des boucles.

Q17. Assez bien traitée dans l'ensemble.

PARTIE IV - GESTION RÉSEAU

Q18. Peu de candidats ont donné les bonnes valeurs. Manifestement, les candidats ont eu des difficultés à s'approprier l'algorithme de Dijkstra en quelques minutes.

Q19. Généralement très bien traitée.

PARTIE V - REQUÊTES SQL

Q20. Question correctement traitée.

Q21. Gros problèmes de syntaxe. Le côté « unicité » de la recherche a souvent été mal vu.

Q22. Gros problèmes de syntaxe.

4/ CONCLUSION

Certaines notions du nouveau programme n'ont pas été bien assimilées par certains candidats. Le sujet et le barème ont permis de classer correctement les candidats.