



## CONCOURS COMMUN INP – SESSION 2022 RAPPORT DE L'ÉPREUVE ÉCRITE INFORMATIQUE

### 1/ PRÉSENTATION DU SUJET

Le sujet abordait la représentation de données géolocalisées sur une carte numérique et s'appuyait sur les programmes de première et deuxième années. Il était constitué de sept parties pouvant être traitées de manière indépendante et de trois annexes.

La difficulté de ce sujet résidait dans l'appropriation de la documentation. Il fallait donc bien la lire et l'exploiter.

### 2/ REMARQUES GÉNÉRALES

#### Erreurs les plus fréquentes des candidats

Les candidats sont souvent trop approximatifs avec le langage Python, ci-dessous les points (par thème) trop souvent mal connus, voire étrangers.

Les variables :

- mettre des noms simples mais explicites (y compris dans les boucles de parcours) ;
- respecter les noms de variables, de fonction, ..., d'une documentation.

Les chaînes de caractères :

- la comparaison de deux chaînes, ordre lexicographique ;
- le « slicing » (extraction par tranches) bien que rappelé en annexe ;
- le caractère non mutable de la chaîne de caractères ;
- itération sur une chaîne de caractères : passage souvent par un lourd `range(len(...))`.

Les opérateurs logiques et les booléens :

- AND remplacé par `&` ou `&&` ;
- `==` remplacé par `=` ;
- `<>` pour `!=` ;
- les booléens `True- False` sont parfois oubliés (ne pas écrire vrai ou faux) ;
- méconnaissance des mots clefs `not` et `in` (test d'appartenance) ;
- il serait sans doute utile de rappeler-expliquer les règles en python sur la valeur booléenne d'une variable (les nombres valent `True` sauf 0 qui vaut `False`, une liste, tuple, string, tout ce qui est vide vaut `False`, tout ce qui n'est pas vide vaut `True`).

Les tuples :

- l'extraction des données d'un tuple ;
- accès à un élément d'un tuple par la place ;
- caractère non mutable du tuple ;

- addition de deux tuples  $t1+t2$  renvoie un tuple qui est la concaténation des deux tuples  $t1, t2$  (addition non commutative).

Parcours des itérables (string, listes ...) :

- For x in non maîtrisés ;
- trop souvent l'utilisation `for i in range(len(...))` se substitue au `for x in ...`, ce qui alourdit le code ;
- les compréhensions de listes ont rarement été utilisées mais quand elles le sont, elles sont parfaitement maîtrisées ;

Les fonctions :

- le « return » n'est pas compris par une quantité trop importante de candidats. En particulier, nombreux considèrent qu'un « print » a le même rôle qu'un « return » ;
- respecter le prototype d'une fonction lorsque celle-ci est donnée, en particulier le type des données retournées et le type de données à lui passer ;
- revoir la récursivité et les règles d'arrêt pour éviter une boucle infinie d'appels ;
- la documentation donne un certain nombre de fonctions et trop souvent les candidats oublient de récupérer le résultat dans une variable ou ne respectent pas les types demandés lors du passage des paramètres.

SQL :

- les jointures ne sont pas souvent maîtrisées ;
- respect de la syntaxe.

Mathématiques :

- bases sur les rectangles oubliées ;
- milieu d'un bipoint non connu ;
- conversion degré – minutes – secondes en décimale.

Algorithmes (cours non connu) :

- algorithme de tri non connu ;
- complexité des algorithmes.

Points généraux et conseils :

- lire une documentation, ici en Français (donc sans difficultés majeure a priori), fait partie des techniques de base de la programmation (accès à des modules, fonctions, ..., données définies ou écrites par d'autres) et de l'algorithmique (compréhension des Entrées -Sorties des fonctions données) ;
- souvent un code python clair, avec respect des règles de syntaxe, y compris celles de la documentation sont le signe d'une copie correcte à excellente ;
- a contrario un code flou, redondant, ne respectant pas les règles Python, est souvent synonyme de copies de piètre qualité ;
- respecter l'ordre des questions (quitte à laisser des blancs) ;
- respecter les règles de couleurs de stylos (quelques rares copies sont encore en bleu très clair) ;
- encadrer l'essentiel si nécessaire et aérer son code ;
- en cas d'erreur, ne pas hésiter à barrer (proprement, pas en barbouillant) la totalité de la réponse puis, réécrire le code proprement en dessous, ce qui prendra moins de temps que de barrer-écrire-barrer ...

## Bilan

Beaucoup de bonnes copies qui montrent que :

- le cours est connu (Complexité, SQL, ...)
- le langage Python est bien connu ;
- la partie algorithmique est connue et maîtrisée.

A contrario, trop de candidats :

- ne connaissent pas leur cours (langage Python et / ou SQL, algorithmes) ;
- ont des difficultés pour lire et s'approprier une documentation en Français sur Python.

## 3/ REMARQUES SPÉCIFIQUES

### PARTIE I

**Q1.** Trop peu de candidats arrivent au bon résultat, la comparaison entre les différentes chaînes de caractères n'est pas comprise par une majorité. Étrangement, certains n'hésitent pas à remettre en cause le sujet au lieu de s'interroger sur leurs propres connaissances des chaînes de caractères.

**Q2.** En général bien traitée.

**Q3.** Le Slicing n'est pas connu par beaucoup de candidats. Nombreux sont ceux qui utilisent une boucle pour tenter de résoudre cette question simple.

**Q4.** En général bien traitée.

### PARTIE II

**Q5.** Certains candidats rendent un tuple à la place de l'expression demandée. La conversion des données a été compliquée dans de trop nombreux cas.

**Q6.** L'utilisation de la fonction `convertTodecimale` est souvent oubliée. L'orientation (N-S et W-E) pour trouver le signe de la latitude et la longitude n'a pas été comprise.

**Q7.** L'appel à la fonction `open image()` est trop souvent oubliée ainsi que l'initialisation de la liste que l'on veut retourner.

**Q8. et Q9.** RAS

### PARTIE III

**Q10.** Peu de candidats font appel à la méthode `transferteTocarte()`. Certains oublient de faire référence à la fonction `open image()`, l'initialisation d'une carte a été souvent oubliée. Beaucoup, se contentent de « remplir » la carte mais oublient de la renvoyer : celle-ci n'est alors plus accessible...

**Q11.** Peu de candidats sont arrivés à résoudre correctement cette question.

**Q12.** Beaucoup d'erreurs sur cette question, on est sur des problèmes simples de géométrie.

**Q13.** La moitié des candidats ne savent pas comment déterminer le milieu de deux points en sachant leurs coordonnées.

**Q14.** Question généralement bien traitée quand elle a été abordée.

**Q15.** Cette question fait appel à de nombreuses fonctions créées préalablement et peut se traiter sans pour autant avoir été capable de les écrire. Pour la traiter convenablement, il faut avoir compris le rôle de chaque fonction. Ainsi, elle a été, soit correctement traitée, soit traitée en utilisant – partiellement parfois – les fonctions mais dans un ordre hasardeux.

#### **PARTIE IV**

**Q16.** Même remarque que **Q15**.

**Q17.** Dans l'ensemble, question assez bien traitée, toutefois, la liste attendue pour la variable locations est oubliée ou non définie. Certains candidats ont oublié de fermer la ligne polygonale (5 éléments dans la liste et non pas quatre).

#### **PARTIE V**

**Q18.** Question globalement mal traitée. Les notions de répertoire et de fichiers semblent être source de confusion chez de nombreux candidats.

**Q19.** Ici, la difficulté réside dans le fait qu'on manipule des chaînes de caractères, la concaténation, et une fonction qui prend en entrée une chaîne de caractères et qui retourne une liste de données. La concaténation de chaînes semble peu maîtrisée et donc trop peu de candidats ont compris comment utiliser la fonction `os.listdir`.

**Q20.** Question difficile qui nécessite la maîtrise de tout ou partie de :

- la notion de fichier et de répertoire ;
- la fonction `os.listdir` ;
- la concaténation ;
- la récursivité et trouver son test d'arrêt.

Très rarement traitée, mais à noter que lorsqu'elle l'a été, le code était parfait.

#### **PARTIE VI**

**Q21.** L'étude du tri est assez bien comprise, toutefois le cours n'est pas assez connu (nom de l'algorithme, complexité). Le rôle du `break` n'est pas bien compris.

**Q22.** Question assez peu traitée. Les candidats qui l'ont traitée l'ont correctement appréhendée, ce qui montre que ces candidats ont compris l'enchaînement des fonctions utilisées et le résultat obtenu.

#### **PARTIE VII**

**Q23. - Q26.** La partie sur SQL a permis à beaucoup de candidats de récolter facilement des points. Toutefois, on note beaucoup d'erreurs de syntaxe (`when` au lieu de `where`) et l'écriture d'une jointure est trop souvent problématique.

La mise en application des requêtes dans le code Python est quant à lui compliqué : une requête SQL semble donc rester un objet abstrait sans utilité autre que de ramasser des points...

## 4/ CONCLUSION

La maîtrise du langage Python s'est largement améliorée au fil des années. Ainsi, on voit d'excellentes copies dans lesquelles les compréhensions de listes et la notion de fichier (open...) sont maîtrisées.

Cependant, il est très étonnant de voir que le « print » se substitue régulièrement au « return » dans une fonction. Souvent utilisé pour faire des tests, le « print » devient ainsi le but ultime du code !

L'utilisation plus systématique des « assert » dans le cadre du débogage est sans doute la piste la plus propice pour que la fonction « print » retrouve son rôle. Cependant, la réalité des copies montre que les candidats qui utilisent à mauvais escient le « print » sont aussi ceux pour qui Python présente le plus de difficultés.

L'utilisation du langage SQL reste encore approximative, et donc une mise en application des requêtes ne peut-être que plus compliquée.

Point essentiel à travailler : les chaînes de caractères, leur maîtrise est loin d'être acquise pour un bon nombre de candidats. Ainsi, la technique du saucissonnage (slicing), qui permet de récupérer tout ou partie de celle-ci dans l'ordre que l'on souhaite, est à améliorer.

Conformément au programme, une date peut être représentée par une chaîne de caractères, l'ordre lexicographique et le slicing permettent la comparaison de dates sur des jours, mois ..., ce point est sans doute à travailler pour beaucoup de candidats.

Concernant l'algorithmique, de bonnes à excellentes copies se démarquent par la clarté des explications, la maîtrise du cours, et la capacité à lire une documentation, se l'approprier et l'exploiter. L'importation de modules doit être l'occasion pour les candidats de lire la documentation associée et d'utiliser les fonctions en respectant les entrées-sorties (type de données attendues, respect de la position et/ou du nommage).