

# Informatique

## Présentation du sujet

Le sujet étudie les circuits de formule 1. La première partie traite une modélisation discrète, assez simple, d'un circuit automobile. La seconde partie est une modélisation continue, plus complexe. La troisième partie étudie le parcours d'une voiture sur les circuits.

Enfin, la quatrième partie évalue les candidats sur les bases de données.

## Analyse globale des résultats

Chacune des parties du sujet est progressive, ce qui a permis d'étaler les notes. Toutes les parties du sujet ont été traitées par les candidats, 124 candidats ont abordé l'intégralité des questions du sujet et 13 d'entre eux n'ont eu zéro à aucune question.

Les candidats ont pu montrer leurs compétences en informatique. Cette année, le sujet a mis l'accent sur les compétences en programmation.

Le jury est globalement satisfait du niveau en informatique atteint par les candidats et encourage les futurs candidats à travailler cette matière importante dans le cursus d'un ingénieur.

## Commentaires sur les réponses apportées et conseils aux futurs candidats

### Programmation en Python

Le sujet comporte un grand nombre de questions de programmation en Python où il est demandé aux candidats d'écrire une fonction satisfaisant une certaine spécification. Cette spécification contient plusieurs exigences (type d'entrée, type de sortie, ce que doit calculer la fonction, ...). Les candidats sont invités à faire attention à toutes ces exigences. Idéalement, la fonction écrite par le candidat les satisfait toutes.

Cette année, le jury a remarqué que les types d'entrée et de sortie n'ont pas toujours été respectés par les candidats. Néanmoins, le jury constate avec satisfaction que les candidats sont moins gênés par des détails techniques (par exemple les bornes correctes des boucles `for`) et font preuve d'une certaine aisance en programmation.

### SQL

Le sujet comporte cinq questions sur les bases de données. Deux d'entre elles (**Q 27.**, **Q 28.**) demandent d'écrire une requête SQL. Une demande de comprendre une requête déjà écrite (**Q 29.**). Le jury est globalement satisfait de la maîtrise de SQL par les candidats.

Les deux autres questions (**Q 25.** et **Q 26.**) se posent en amont de la construction de la base de données : quelle clef primaire choisir ? Quelle va être la taille de ma base de données ? Il est important de bien comprendre qu'on ne peut pas utiliser la base de données pour répondre à des questions qui se sont posées en amont de sa création.

### Manipulation des booléens

La manipulation des booléens est un point difficile, sur plusieurs niveaux.

Articuler correctement les conditions, éviter les imbrications de « if », demande de travailler la logique.

En outre, les candidats devraient avoir conscience que les booléens sont des valeurs comme les autres, qu'ils peuvent être manipulés comme n'importe quel autre type de valeurs (entiers, flottants, etc.). Certains codes trahissent une mauvaise compréhension du rôle des booléens, par exemple des codes de la forme suivante :

```
if condition == True :
    return True
else :
    return False
```

### Questions faisant appel à des connaissances d'autres disciplines

Plusieurs questions font appel à des connaissances de mathématiques ou de physique. Les candidats doivent être capable de mobiliser les connaissances qu'ils ont acquis dans d'autres disciplines. Il est important de garder un recul critique sur les résultats des applications numériques.

### Read the manual

Plusieurs questions (**Q 9.**, **Q 12.**) demandent d'utiliser des fonctions dont la description est donnée dans le sujet. Le jury encourage les candidats à bien lire et assimiler les descriptions de ces fonctions. Ces questions ont été assez bien traitées, même si souvent, quelques détails techniques étaient négligés.

### Lisibilité

Le jury insiste sur l'effort que les candidats doivent faire pour être lisibles, Si certaines copies étaient parfaitement claires et compréhensibles, d'autres étaient plus difficiles à déchiffrer, ce qui a pu créer des ambiguïtés dans le code Python (par exemple, dans certaines copies, il était difficile de différencier certains noms de variables).

Les candidats sont invités à faire particulièrement attention à la lisibilité de leur code. Cette lisibilité peut souvent être améliorée facilement en évitant certains noms de variables (comme `o`), et en évitant de donner des noms trop semblables à des variables différentes.

### Remarques particulières sur certaines questions

**Q 2.** Cette question évalue la capacité des candidats à exécuter à la main une fonction pour en déterminer le résultat. C'est une compétence importante, utile à la compréhension de la programmation. Le jury est peu satisfait des réponses à cette question. 96 % des candidats ont traité cette question, 62 % des candidats ont eu une réponse au moins partiellement correcte, seulement 25 % des candidats ont traité parfaitement cette question.

**Q 5.** Cette question demande de combiner plusieurs conditions, en particulier, que le nombre de virages à gauche soit égal à  $0 \bmod 4$  n'est pas une condition suffisante. Il faut gérer à la fois une condition sur la direction, et une condition sur la position. Ceci a présenté une difficulté à 16 % des candidats qui n'ont pas traité ou ont eu zéro à cette question.

**Q 21.** et **Q 22.** La fonction `temps_droite` demandée à la **Q 21.** doit renvoyer un couple de flottants. Cette exigence a perturbé certains candidats qui n'ont renvoyé qu'une des deux valeurs demandées ou qui ont utilisé `temps_droite` à la **Q 22.** comme si elle renvoyait un unique flottant.

**Q 25.** Dans cette question, il est demandé de justifier pourquoi la base de données a été construite, Les arguments utilisant la structure de la base de données (comme la présence du champ `pi_id` dans d'autres

tables) ne sont donc pas pertinent. Les candidats doivent avoir conscience que des choix ont été faits lors de la création de la base de données, que d'autres choix auraient pu être fait et doivent être capable d'identifier ces choix et de discuter de leur pertinence. 62 % des candidats ( 72 % des candidats ayant abordé cette question) l'ont parfaitement traitée.

**Q 26.** Cette question ne demande pas de requête, mais demande d'estimer la taille des données que la base de données devra traiter. L'objectif est d'estimer la taille que va prendre la base de données avant de la construire.

Il est aussi attendu des candidats d'avoir un regard critique sur les valeurs numériques qu'ils obtiennent. Si le candidat trouve qu'il y a moins de 100 lignes à traiter, il peut s'interroger sur la pertinence d'utiliser une base de données.

Cette question est la moins bien réussie de la partie IV. Seuls 14 % des candidats l'ont parfaitement traitée (27 % des candidats ont eu des points sur cette question).

## Conclusion

Les résultats des étudiants sont globalement satisfaisants. Le recul vis-à-vis du problème posé, la compréhension des objets manipulés, la clarté et la concision de l'expression sont autant d'atouts qui serviront les candidats en informatique et, plus généralement, dans leur carrière d'ingénieur.