Informatique

Présentation du sujet

Le sujet d'informatique 2022, commun à toutes les filières, proposait de modéliser des circuits et des courses de F1 sur ces circuits. Quatre parties composaient le sujet décliné en 29 questions.

— La première partie, avec 7 questions, proposait une modélisation d'un circuit de course, qualifiée de sommaire. Elle était l'occasion de définir des premières fonctions en Python.

— La deuxième partie, avec 5 questions, prolongeait la précédente en proposant une représentation plus réaliste d'un circuit. Après une contextualisation du problème, cette partie était encore l'occasion d'écrire des fonctions en Python.

— La troisième partie, avec 12 questions, comportait deux sous-parties III-A et III-B. La III-A préparait certains résultats exploités dans la III-B. Elle faisait largement appel aux connaissances du cours de physique. Une mise en œuvre informatique suivait avec l'écriture de fonctions en Python requérant une certaine prise de recul par rapport au sujet et une capacité à articuler toutes les informations des questions précédentes.

— La dernière partie, avec 5 questions, visait essentiellement à écrire des requêtes en langage SQL.

Analyse globale des résultats

La progressivité et la nature des questions ont permis au jury d'évaluer la qualité et le niveau de compétences de chaque candidat en matière de programmation Python et SQL. Le jury observe une baisse des résultats obtenus par rapport à ceux des dernières sessions. Si les bases de l'algorithmique et de la programmation élémentaire dans les langages Python et SQL semblent maitrisées, l'écriture de programmes nécessitant une réflexion amont sur leur organisation et leur structuration est très souvent problématique. Ce qui, outre un niveau global assez faible en informatique, révèle également un manque de pratique du codage. Il serait souhaitable que les candidats mesurent l'importance de leur formation initiale en informatique pour la suite de leurs études.

Un barème précis, construit de manière à évaluer toutes les compétences attendues pour une telle épreuve, a permis une correction aussi équitable et homogène que possible. Quelques rares copies sont de très bonne facture. D'autres, trop nombreuses pour une telle épreuve, révèlent des niveaux très bas en informatique. Quelques candidats ont visiblement négligé la formation.

La mise en place de points malus attribués suivant un protocole clair et précis par le concours a permis de sanctionner les copies ne respectant pas certaines normes de présentation. Le jury attire l'attention des candidats sur la nécessité de produire un travail lisible et respectueux de la langue. L'expression écrite doit être claire. L'identification des questions et la mise en évidence de résultats sont un minimum dans une copie de concours. Enfin, la présentation doit être soignée ; en particulier, le jury déplore un trop grand nombre de ratures qui n'ont pas leur place à ce niveau de formation.

Questions traitées

% de questions traitées	de 0 à 25%	de 25% à 50%	de 50% à 75%	de 75% à 100%
% de copies	1%	13%	49%	37%

Copies ayant obtenu un certain nombre de points

	Partie I	Partie II	Partie III-A	Partie III-B	Partie IV
Au moins 1 point	99%	93%	97%	44%	97%
Au moins la moitié des points	34%	4%	51%	1%	33%

Commentaires sur les réponses apportées et conseils aux futurs candidats

Le jury souhaite attirer l'attention des futurs candidats sur les points suivants.

— La notion de *type*, essentielle en informatique doit prise en compte lors de la manipulation des objets. Les candidats doivent s'interroger sur la pertinence et les limites de certaines opérations effectuées sur ou entre objets de même type. Par ailleurs, le type d'un résultat renvoyé par une fonction doit être respecté.

— Certaines *boucles* utilisent des indices, d'autres itèrent directement sur la structure. Il y a parfois confusion entre ces deux modes des parcours. Il convient d'être plus rigoureux.

— Les *bornes* des boucles doivent être bien comprises. Une instruction de la forme `for i in range(n)` indique que la variable `i` prend ses valeurs entre `0` et `n-1`. Il y a souvent une erreur sur la valeur de la borne supérieure.

— Certains programmes révèlent une compréhension partielle des *booléens* et des *instructions conditionnelles*, comme `if isinstance(e, int) == True` ou encore :

```
if e > 0 : return True
else : return False
```

On attend d'un candidat aux concours aux grandes écoles qu'il comprenne le sens d'une expression de la forme `if isinstance(e, int)` ou `return e > 0` et qu'il sache la réinvestir à bon escient.

— Les *requêtes SQL* doivent faire l'objet d'une attention particulière. Si les questions sur les bases de données sont quasi-systématiquement abordées, peu de candidats obtiennent tous les points en raison d'une syntaxe SQL parfois approximative et d'une maitrise insuffisante de la notion de jointure. En particulier, certaines jointures sont faites à l'aide d'un produit cartésien suivi d'une sélection. L'utilisation de `JOIN ... ON` est à privilégier.

— Les programmes sont globalement syntaxiquement corrects et lisibles. Leur lecture révèle toutefois une écriture au fil de l'eau. Préalablement à l'écriture d'une fonction, il serait souhaitable que chaque candidat s'interroge sur l'*organisation* et la *structure logiques des programmes* qu'il propose.

— Les *commentaires* sont nécessaires mais il convient de ne pas en abuser et d'éviter les paraphrases. Un commentaire n'a de sens que s'il apporte une information utile et pertinente. Les *docstrings* (documentation placée immédiatement après la définition d'une fonction) sont toujours utiles en pratique mais généralement pas attendues sur une copie dans le cadre d'une épreuve de concours en temps limité.

— Le sujet propose une *annexe* détaillant la syntaxe et le rôle de fonctions spécifiques. Certains candidats semblent ne pas avoir lu cette annexe. Par exemple, la fonction `EXTRACT` nécessaire à l'écriture d'une requête n'a été utilisée que par très peu de candidats. Le jury attire l'attention sur la nécessité de lire le sujet dans son ensemble et de savoir extraire les informations des annexes pour répondre efficacement à certaines questions.

– Une réponse à une question de programmation ou de calcul attend un minimum d'explication sous la forme d'une courte introduction. Sans explication, sans argumentation, une réponse est incomplète.

– Un programme incomplet ne fonctionne pas. Beaucoup de candidats débutent l'écriture d'un programme sans le terminer, espérant glaner quelques points. C'est malheureusement un mauvais calcul !

Sont présentées ci-dessous quelques remarques relatives aux question du sujet.

Modélisation sommaire d'un circuit

Q1. Cette question comportait un test de la forme `if e == "A": res += d`. Un certain nombre de candidats ajoute l'alternative inutile `else: res = res`. Il convient de s'interroger sur le sens des instructions conditionnelles.

Q2. Les arguments invoqués pour répondre à cette question sont souvent maladroits.

Q3. Les réponses à cette question sont globalement satisfaisantes.

Q4. Il y a parfois confusion entre les parcours de listes avec un `in` et un `range`. Par ailleurs, comme cela est écrit en remarque générale, attention au débordement. Par ailleurs, cette question requérait l'appel à la fonction définie à la question 1, souvent oubliée. Enfin, il y a parfois confusion entre `False` et `True`.

Q5. Cette question n'a pas été réussie. Plusieurs tests devaient mener à l'établissement du résultat. Ces tests n'ont pas toujours été bien écrits.

Q6. Comme la question précédente, celle-ci nécessitait l'écriture de tests successifs faisant évoluer certaines variables définies localement. Elle fut peu réussie.

Q7. En revanche, cette question a été bien traitée.

Modélisation plus réaliste d'un circuit

Q8. La question est abordée par la majorité des candidats mais est soit maladroitement traitée, soit incomplète ; souvent les deux !

Q9. Le facteur d'échelle est souvent absent.

Q10. Trop peu de candidats pensent à convertir l'angle en radian. La notion de matrice de rotation n'est pas toujours connue.

Q11. Cette question exigeait une compréhension du texte et une capacité à expliquer. Les résultats sont très inégaux et révèlent parfois des difficultés d'expression écrite.

Q12. Cette seule question sur la complexité a été très largement non traitée.

Le parcours d'une voiture

Q13. - Q16. Ces questions faisaient appel à des connaissances de physique élémentaire. Dans l'ensemble, les réponses sont satisfaisantes mais quelques copies éprouvent des difficultés à y répondre.

Q17. - Q19. Ces questions nécessitaient de faire appel aux résultats des quatre questions précédentes. Elles ont souvent posé des difficultés, les réponses étant souvent soit incomplètes, soit incorrectes.

Q20. - Q22. Ces questions devaient mener à l'écriture de fonctions utilisant les résultats des questions 13 à 19. La longueur des programmes à écrire, la multiplicité des conditions et des éléments à prendre en compte a certainement conduit de nombreux candidats à sauter ces questions, à ce moment de l'épreuve. En conséquence, ces questions plus exigeantes d'un point de vue informatique ont été très peu réussies quand elles ont été abordées.

Notons que certaines propositions ne respectent le cahier des charges. Par exemple, à la question 21, la fonction devait renvoyer deux valeurs, temps et vitesse. La vitesse est souvent oubliée.

Q23. Bien que peu souvent abordée, cette question a reçu des réponses diverses, parfois avec des justifications pertinentes. D'autres fois, aucune explication n'est fournie. Nous le répétons : il convient d'argumenter, d'expliquer, de justifier les choix algorithmiques et les choix de codages.

Q24. Question peu traitée, mais trop souvent `O(n c)` au lieu de `O(n len(c))`.

IV Gestion des résultats

Q25. Globalement comprise et bien traitée.

Q26. Quelques candidats répondent par une requête SQL. Certains candidats ne semblent pas avoir compris la question qui nécessitait un calcul d'ordre de grandeur.

Q27. Bien réussie dans l'ensemble, l'écriture d'une requête simple avec une jointure semble maitrisée.

Q28. Cela devient plus difficile quand plusieurs jointures sont nécessaires. Sans parler de clauses visant à filtrer les données extraites.

Q29. Cette question est plutôt bien traitée par les candidats qui l'aborde.

Conclusion

? ? ? ? ? ?