



1/ Présentation de l'épreuve

L'épreuve portait sur l'étude d'un système informatique de gestion de déchetteries.

Le sujet était décomposé en 4 parties totalement indépendantes :

- la première partie du sujet portait sur la modélisation du comportement du système informatique lors du passage des usagers sur le pont pour entrer dans la déchetterie ;
- la seconde partie portait sur l'étude de l'authentification de l'utilisateur de la déchetterie ;
- la troisième partie portait sur l'étude du bloc logiciel de traitement du signal intégré au peson ;
- la quatrième partie portait sur l'étude de la base de données « centrale » du système.

2/ Remarques générales

2.1/ Remarques générales sur le sujet

Le sujet proposé valide les compétences du programme :

- analyser et modéliser un problème, une situation ;
- imaginer et concevoir ;
- traduire un algorithme dans un langage de programmation moderne et généraliste ;
- évaluer, contrôler, valider des algorithmes et des programmes.

Partie 1 : cette partie a été simplifiée par rapport au programme réel, programme traitant les entrées et les sorties de la déchetterie par exclusion mutuelle (qui n'est pas au programme de CPGE). Il a été demandé aux candidats de créer un algorithme décrivant le fonctionnement du système lors de l'entrée d'un utilisateur dans la déchetterie.

⇒ Référence au programme : algorithmique et programmation I

Partie 2 : après l'étude du protocole de communication et de l'algorithme de lecture des informations du lecteur RFID, il a été demandé aux candidats de travailler sur le tri des numéros de badge et la recherche d'un badge particulier afin d'autoriser son porteur à entrer dans la déchetterie. Le questionnement portait également sur la terminaison et la complexité de l'algorithme proposé.

⇒ Références au programme :

- algorithmique et programmation I
- algorithmique et programmation II

Partie 3 : cette partie proposait la modélisation du signal réel du peson puis son filtrage par convolution par un filtre gaussien.

⇒ Référence au programme : ingénierie numérique et simulation

Partie 4 : cette partie portant sur les bases de données a permis de rendre compte du traitement des informations liées aux utilisateurs des déchetteries en finissant par une question portant sur la valorisation des déchets, finalité du système.

⇒ Référence au programme : initiation aux bases de données

2.2/ Remarques générales sur la composition des candidats

Globalement le sujet a été traité dans son ensemble par beaucoup de candidats et toutes les questions du sujet ont été traitées, contrairement à l'année précédente. Ainsi, l'étalement des notes de cette épreuve permet de très bien classer les candidats. On note cependant qu'il reste encore une infime partie des candidats ne parvenant à ne traiter qu'une toute petite partie du questionnement proposé.

Nous pouvons encore noter cette année qu'il y a un manque important de rigueur dans beaucoup de copies dans l'écriture des algorithmes. Cela nuit évidemment aux candidats concernés.

La majorité des candidats est dans l'incapacité de prendre du recul sur le problème posé. Des éléments importants permettant de répondre aux questions apparaissent souvent dans le texte ou dans les annexes, mais une grande partie des candidats passe à côté de ces informations.

Enfin, sur 1 064 copies, 5 candidats ont composé en Scilab. Certains de ces candidats ont également écrit quelques réponses en Python, proposant parfois les deux réponses pour une même question. Le contenu de ces copies est en général assez peu développé. Ainsi, **pour la session 2017, seul le langage Python sera utilisé dans l'épreuve d'informatique de la filière TSI.**

3/ Remarques spécifiques

Question 1.

Beaucoup de candidats se contentent de recopier à la suite les fonctions proposées dans l'énoncé sans réellement chercher à comprendre le fonctionnement de la déchetterie. Cette question ne présentait pourtant pas de difficulté particulière.

Question 2.

Cette question ne présentait pas de difficulté particulière.

Le calcul du checksum n'est souvent pas effectué.

Bien qu'il y ait la table de vérité de l'opérateur OU EXclusif dans le sujet, certains traitent la question en utilisant un opérateur ET.

Question 3.

Il s'agissait de commenter un algorithme proposé.

Beaucoup de candidats se contentent d'expliquer chaque mot du code sans réellement prendre de recul et ne cherchent pas à montrer l'utilité de la séquence.

Question 4.

Les candidats devaient ici élaborer un programme en Python ou Scilab, d'après l'algorithme proposé à la question précédente.

Cette question a été souvent bien traitée par les candidats.

Question 5.

Un très grand nombre de candidats ne connaît pas encore le concept de Client/Serveur d'un applicatif type 2 tiers.

Question 6.

Il s'agissait dans cette question de vérifier que les candidats étaient en mesure de valider le fonctionnement d'un algorithme simple en leur faisant compléter, dans le document réponse, les échanges de place des informations à trier.

Cette question ne présentait pas de difficulté particulière, cependant beaucoup de candidats n'ont pas bien su compléter correctement la trace de l'algorithme ou n'ont pas correctement appréhendé la manière de trier.

Question 7.

Question bien traitée par les candidats.

Question 8.

Cette question nécessitait de connaître la complexité des algorithmes. Peu de candidats ont justifié leur réponse.

Les notions de complexité semblent ne pas encore être suffisamment comprises et certaines confusions ont été faites comme par exemple une complexité $O(n^2)$ nommée cubique.

Question 9.

Certains candidats confondent récursivité et récurrence.

Beaucoup n'ont pas compris les précautions à prendre sur les paramètres m et n lors de l'appel de la fonction.

Question 10.

Les correcteurs attendaient un algorithme de recherche d'occurrence non récursif. Cette question a été trop peu traitée.

Peu de candidats ont réellement appréhendé le fonctionnement d'un algorithme de type récursif et en particulier l'utilisation de la mémoire lors de son exécution.

Question 11.

Beaucoup de candidats n'ont pas su lire le spectre du signal échantillonné, mais la plupart connaissent la condition de Nyquist-Shannon.

Question 12.

Cette question avait pour objectif de justifier les paramètres retenus comme étant ceux d'un premier ordre. Cette question a été traitée maladroitement ou mal comprise dans de nombreux cas.

Questions 13 et 14.

A priori, ces questions présentaient peu de difficultés, mais peu de candidats ont été en mesure de donner une réponse complète et certains oublient des paramètres.

Question 15.

Question assez bien réussie par les candidats. Nous pouvons noter quand même que certains candidats recopient textuellement la forme littérale de l'équation sans prendre en compte le langage de programmation, ce qui est regrettable.

Question 16.

Cette question n'a pas été bien traitée : beaucoup de candidats n'ont pas compris qu'ils pouvaient réutiliser les fonctions déjà proposées, `signal()` et `bruitGauss()`, pour le traitement demandé.

Question 17.

Une des trois questions terminales de cette partie, question peu ou mal traitée, car les candidats ont été dans l'impossibilité d'écrire l'équation voulue en Python ou Scilab.

On peut remarquer que certains candidats ne connaissent pas l'opérateur `+=` en Python.

Question 18.

Le calcul de σ a été relativement bien opéré car c'est une recopie de l'équation fournie. Par contre, l'établissement du tableau de la réponse n'a été que très rarement abordé.

Seuls les meilleurs candidats ont réussi.

Question 19.

Question terminale de la partie.

Beaucoup de candidats ont bien vu le lissage et que la donnée représentait bien la masse mesurée. À l'inverse, le temps de retard n'a pas été vu et aucun candidat n'a émis une hypothèse sur la prise en compte de ce retard dans le programme de gestion.

Question 20.

Cette question ne présentait pas de difficulté particulière à part une lecture des documents.

Question 21.

Question traitée par pratiquement tous les candidats avec un bon taux de réussite. Néanmoins, certains candidats ne connaissent pas le terme « clé étrangère », concept pourtant indissociable de l'étude des bases de données relationnelles.

Question 22.

Pas de difficulté particulière pour cette question, un nombre conséquent de candidats a cependant oublié la condition « autorisation='oui' », car tous les utilisateurs présents dans la table de la base de données ne sont pas forcément autorisés à effectuer des dépôts.

Question 23.

Le concept de relation dans l'étude des bases de données relationnelles implique des contraintes référentielles entre tables.

Très peu de candidats ont vu qu'il fallait que la ville de « VILLENEUVE sur YONNE » soit déjà présente dans la table « COMMUNES » en tant qu'entité avant de pouvoir insérer un nouvel utilisateur provenant de cette ville.

Question 24.

Cette question portait sur la mise en place d'une jointure entre les tables « COMMUNES » et « UTILISATEURS ». La question a été relativement bien traitée, cependant le classement alphabétique impliquant la commande « ORDER BY » a souvent été oublié.

Question 25.

Cette question terminale avait pour objectif l'utilisation des fonctions d'agrégation en SQL afin de répondre à la mission du système.

Très peu de candidats ont utilisé les fonctions count() et sum() afin de répondre à la question.

Enfin, quelques points ont été attribués aux candidats sur le soin apporté à leurs copies, dans l'écriture des algorithmes et programmes en Python ou Scilab ainsi que pour l'orthographe.

4/ Conseils aux candidats

Les correcteurs conseillent aux candidats d'avoir une meilleure maîtrise des concepts de base de la programmation (structures algorithmiques, définitions de fonctions, appels de fonctions, fonctions récursives...), une meilleure maîtrise du langage de programmation Python (syntaxe de base) et une maîtrise plus large des concepts utilisés dans les Bases de Données.

Les correcteurs encouragent les candidats à prendre davantage de recul pour traiter les épreuves proposées.

5/ Histogramme

Nombre de copies : 1064

Moyenne : 9,74 / 20

Écart-type : 3,64

