

Concours A2GP/GCN/GIN session 2025

Option MATHS

Option PHYSIQUE & CHIMIE

Option SCIENCES DE L'INGÉNIEUR

Option MATHS, SCIENCES DE L'INGÉNIEUR ET PHYSIQUE

Option CHIMIE

Composition : **Informatique 1**

Durée : **3 Heures**

Consignes pour les candidats

Merci de ne rien marquer sur le sujet.
Pour chaque question de l'épreuve, une seule bonne réponse possible.
Répondez sur la grille séparée qui comporte 18 questions (Q01 à Q18).
Seules les grilles correctement remplies seront corrigées.
L'option *Autre réponse* des QCM vous permettra d'inscrire la bonne réponse ne figurant pas parmi les réponses proposées.
L'exercice 3 comporte 7 questions (Q19 à Q25)

EXERCICE 1 : Représentation des nombres et précision (05 points)

La sonde *Inp-7*, en mission d'étude des éruptions solaires, subit une tempête magnétique qui corrompt ses transmissions. Les données scientifiques vitales pour la mission sont altérées. Vous êtes ingénieur-e en chef chargé du diagnostic urgent au niveau du Centre de contrôle de mission *Ccm-7*. Le centre de contrôle de mission *Ccm-7* est la structure qui rassemble les moyens nécessaires à la gestion opérationnelle de la sonde spatiale explorant le système solaire. Votre rôle au niveau du *Ccm-7* est de gérer non seulement le déroulement des missions spatiales du décollage jusqu'à l'atterrissage ou à la fin de la mission, mais aussi de surveiller tous les aspects de la mission en utilisant les télémesures. Les spécifications techniques de l'architecture embarquée de la sonde en mission d'étude sont les suivantes :

- Entiers signés : Complément à 2 sur 8/16 bits
- Flottants : IEEE 754 simple précision (32 bits)
- Protocole : Trames de 64 octets avec CRC-16-CCITT (algorithme de Contrôle de Redondance Cyclique, utilisé pour détecter les erreurs dans la transmission de données)

La structure des données de télémesure se présente comme suit :

```
Telemesure {  
    Statut; // Plage de représentation  
    Horloge; // Modulo 65536 (16 bits non signés)  
    Flux_UV; // Format IEEE 754  
    CRC; // Polynôme 0x1021  
}
```

Lors de la communication entre *Inp-7* et *Ccm-7*, la dernière trame reçue par le *Ccm-7* est décrite par la structure ci-dessous :

```
Statut   : 0xB6 → 10110110 (binaire)  
Horloge  : 0x007B → 00000000 01111011  
Flux_UV  : 0xC3DA0000 → 11000011 11011010 00000000 00000000 (IEEE 754)  
CRC      : 0xA93D → 10101001 00111101
```

Q01. La méthode du complément à deux (*two's complement*) est utilisée en informatique pour représenter les nombres négatifs. Considérons un nombre entier $-k$ (k positif). Le principe de cette méthode est le suivant :

- A Utiliser directement l'écriture binaire de $-k$
- B Prendre le binaire de k , ajouter 1 puis inverser les bits
- C Prendre le binaire de k , ajouter le binaire de 2 puis inverser les bits
- D Multiplier par -1 la représentation binaire de k
- E Inverser les bits du binaire de k puis ajouter 1
- F Inverser les bits du binaire de k puis ajouter le binaire de 2
- G Autre réponse

Q02. Conversion d'entier signé : Que vaut en décimal signé, la représentation $0xB6$ issue de la dernière trame reçue par *Ccm-7* ?

- A -34
- B -44
- C -54
- D -64
- E -74
- F -84
- G Autre réponse

Q03. Plage de représentation : quelle est la valeur minimale d'un entier 8 bits signé ?

- A -32767
- B -32768
- C -127
- D -128
- E -7
- F -8
- G Autre réponse

Q04. Si la sonde stocke 150 dans un entier 8 bits signé, que se passe-t-il ?:

- A Une saturation à 127
- B Une erreur système au niveau de la sonde *Inp-7*
- C Une saturation à 150
- D Un débordement en complément à 2 d'une valeur de -106
- E Autre réponse

Q05. Considérons le flux ultraviolets naturels $0xC3DA0000$ reçu par le *Ccm-7*. Quelle est la valeur décimale de l'exposant IEEE 754 de ce nombre ?

- A 7
- B 8
- C 9
- D 10
- E 11
- F 12
- G Autre réponse

Q06. Considérons le flux ultraviolets naturels $0xC3DA0000$ reçu par le *Ccm-7*. Quelle est la valeur décimale de la mantisse IEEE 754 de ce nombre ?

- A 0,75
- B 0,703125
- C 0,715
- D 0,725
- E 0,125
- F Autre réponse

Q07. Considérons le flux ultraviolets naturels $0xC3DA0000$ reçu par le *Ccm-7*. Quelle est la valeur de la partie entière IEEE 754 de ce nombre ?

- A -436
- B -437
- C -438
- D -439
- E -440
- F Autre réponse

Q08. Pourquoi 12,4 ne peut-il être stocké exactement en IEEE 754 ?

- A Représentation binaire périodique
- B Limite de l'exposant
- C Mantisse trop petite
- D Erreur de CRC
- E Autre réponse

Q09. Pourquoi **10**308** ne cause-t-il pas d'erreur en Python ?

- | | |
|---|------------------------------------|
| A | Conversion automatique en flottant |
| B | Précision arbitraire des entiers |
| C | Optimisation du compilateur |
| D | Vérification CRC |
| E | Autre réponse |

Q10. Sachant que le flux UV lu est différent de la valeur attendue qui est de $-26,5$, identifier l'origine de l'erreur.

- | | |
|---|---------------------------|
| A | Bit flip dans la mantisse |
| B | Inversion de signe |
| C | Bit flip dans l'exposant |
| D | Débordement entier |
| E | Autre réponse |

EXERCICE 2 : Programmation Python (08 points)

Vous êtes ingénieur-e dans un établissement scolaire de Yamoussoukro (Côte d'Ivoire). Votre rôle est d'automatiser des tâches critiques afin d'améliorer l'efficacité des services de cette école en utilisant le langage de programmation Python. Pour ce faire, chaque question ci-dessous correspond à une fonctionnalité critique que vous devez implémenter pour atteindre votre objectif.

Pour chaque fonction proposée ci-dessous, sauf l'une des structures A, B, C, D, E ou F associée à cette fonction est correcte.

Q11. Recherche d'un élève en urgence : La direction doit vérifier si *Louise Dupont* est bien inscrite en Terminale G2. La liste des élèves est stockée sous forme de tuples (nom, classe). Quelle fonction donne la réponse exacte ?

<pre>elevés = [("Paul Leroy", "TermG1"), ("Louise Dupont", "TermG2"), ("Amine Zaid", "TermG2")] def est_inscrit(nom, classe):</pre>	
A	<pre>return classe in [e[1] for e in élevés]</pre>
B	<pre>return (nom, classe) in élevés</pre>
C	<pre>return (classe, nom) in élevés</pre>
D	<pre>for e in élevés: if e[1] == nom: return True return False</pre>
E	<pre>return nom in élevés</pre>
F	<pre>for e in élevés: if e[0] == nom: return True return False</pre>

Q12. Identifier la meilleure note : Le professeur de Maths a stocké les notes dans un dictionnaire. Trouvez la note maximale de *Sophie*.

<pre>notes = {"Sophie": [15, 17], "Lucas": [12, 14], "Inès": [18, 19]} def meilleure_note(nom):</pre>	
A	<code>return sum(notes[nom]) / len(notes[nom])</code>
B	<code>return max(notes["notes"])</code>
C	<code>return max(notes[nom for nom in notes])</code>
D	<code>return max(notes.values())</code>
E	<code>return sorted(notes[nom])[-1]</code>
F	<code>return max([n for sublist in notes.values() for n in sublist])</code>

Q13. Compter les spécialités choisies : Déterminer le nombre d'élèves par spécialité ? Les données sont dans une liste désordonnée.

<pre>specialites = ["Physique", "Maths", "Physique", "SVT", "Maths", "Maths"] def compte_choix():</pre>	
A	<code>return {s: len(s) for s in specialites}</code>
B	<code>return dict(zip(specialites, [1]*len(specialites)))</code>
C	<code>d = {} return d</code>
D	<code>d = {} for s in specialites: d[s] = d.get(s, 0) + 1 return d</code>
E	<code>return {s: specialites.count(s) for s in set(specialites)} if False else None</code>
F	<code>return {s: i for i, s in enumerate(specialites)}</code>

Q14. Détecter un mot-clé caché : Un message codé contient le mot "*Réunion*". Comment vérifier sa présence dans ce message sans utiliser `find` ?

<pre>texte = "LaRéunionDesProfesseursEstAnnulée" def detecte_mot():</pre>	
A	<code>for c in texte: if c == "Réunion": return True return False</code>
B	<code>return mot in texte.split("Réunion")</code>
C	<code>mot = "Réunion" for i in range(len(texte) - len(mot) + 1): if texte[i:i+len(mot)].lower() == mot: return True return False</code>

D	<code>return "Réunion" in texte</code>
E	<code>return [c for c in mot] in texte</code>
F	<code>return set(mot).issubset(set(texte))</code>

Q15. Éviter les chevauchements de cours : Le proviseur veut planifier le maximum de cours dans une salle sans chevauchement. Quel algorithme utiliser?

NB : Un cours long (ex: 9h-12h) pourrait bloquer la salle et empêcher deux cours courts (ex: 9h-10h et 10h30-11h30). De plus, un cours commençant tôt mais finissant tard (ex: 8h-18h) rend la salle indisponible toute la journée.

("Histoire", 9, 10) signifie : un cours d'Histoire de 9h à 10h

	<code>cours = [("Histoire", 9, 10), ("Maths", 9, 11), ("SVT", 12, 13)]</code> <code>def planifie():</code>
A	<code>cours.sort(key=lambda x: x[2]-x[1], reverse=True)</code> <code>return cours[:2]</code>
B	<code>import random</code> <code>return random.sample(cours, k=2)</code>
C	<code>cours.sort(key=lambda x: x[2])</code> <code>planning = [cours[0]]</code> <code>for c in cours[1:]:</code> <code>if c[1] >= planning[-1][2]:</code> <code>planning.append(c)</code> <code>return planning</code>
D	<code>cours.sort(key=lambda x: x[0])</code> <code>return cours</code>
E	<code>return [c for c in cours if c[2] < 10]</code>
F	<code>planning = []</code> <code>for c in cours:</code> <code>if all(c[1] >= other[2] for other in planning):</code> <code>planning.append(c)</code> <code>return planning</code>

Q16. Gérer la caisse de la cantine : La cantine doit rendre 68 francs avec des pièces [20, 10, 5, 2, 1]. Laquelle des méthodes est optimale ?

	<code>def rendu_monnaie(montant):</code> <code>pieces = [20, 10, 5, 2, 1]</code> <code>res = []</code>
A	<code>return [10] * (montant // 10)</code>
B	<code>for p in pieces:</code> <code>while montant >= p:</code> <code>res.append(p)</code> <code>montant -= p</code> <code>return res</code>
C	<code>import random</code> <code>return random.choices(pieces, k=montant)</code>
D	<code>return [20, 20, 20, 5, 2, 1] if montant == 68 else []</code>

E	<pre>pieces = [1, 2, 5, 10, 20] for p in pieces: res += [p] * (montant // p) montant %= p return res</pre>
F	<pre>return [25, 25, 18]</pre>

Q17. Trouver les jumeaux académiques : Deux élèves ont-ils exactement les mêmes notes en maths et physique ?

NB: "Pierre": (15, 14) signifie : Pierre a 15 en Maths et 14 en Physique

<pre>resultats = {"Pierre": (15, 14), "Lea": (14, 15), "Emma": (15, 14)} def jumeaux_academiques():</pre>	
A	<pre> eleves = list(resultats.keys()) for i in range(len(eleves)): for j in range(i+1, len(eleves)): if resultats[eleves[i]] == resultats[eleves[j]]: return (eleves[i], eleves[j]) return None</pre>
B	<pre> return (eleves[0], eleves[1])</pre>
C	<pre> inversé = {} for nom, notes in resultats.items(): inversé[notes] = nom return inversé</pre>
D	<pre> return [nom for nom in resultats if nom == "Pierre"]</pre>
E	<pre> for nom in resultats: if resultats[nom] == (15, 14): return nom return None</pre>
F	<pre> return resultats[0]</pre>

Q18. Organiser un tournoi sportif : Il faut apparier toutes les équipes pour des matchs sans répétition.

Exemple de résultat attendu : [('A','B'), ('A','C'), ..., ('C','D')]

<pre>equipes = ["A", "B", "C", "D"] def genere_matches():</pre>	
A	<pre> matchs = [] for i in range(len(equipes)): for j in range(i+1, len(equipes)): matchs.append((equipes[i], equipes[j])) return matchs</pre>
B	<pre> return [(e1, e2) for e1 in equipes for e2 in equipes]</pre>
C	<pre> return [(e1, e2) for e1 in equipes for e2 in equipes if e1 != e2]</pre>
D	<pre> from itertools import permutations return list(permutations(equipes, 2))</pre>
E	<pre> return [(e, e) for e in equipes]</pre>
F	<pre> return []</pre>

EXERCICE 3 : Requêtes SQL (07 points)

TechManufacturing CI est une entreprise d'ingénierie spécialisée dans les projets industriels en Côte d'Ivoire. Elle dispose d'une base de données décrivant l'ensemble de ses activités. Le modèle relationnel de cette base de données est représenté de la manière suivante :

- **Ingenieurs** (*id_ingenieur*, nom, specialite, date_embauche)
- **Projets** (*id_projet*, nom_projet, budget, date_debut, date_fin, #id_chef_ingenieur)
- **Taches** (*id_tache*, description, #id_projet, #id_ingenieur, duree_jours)
- **Fournisseurs** (*id_fournisseur*, nom_fournisseur, pays)
- **Livraisons** (*id_livraison*, #id_fournisseur, #id_projet, date_livraison, cout)

Dans ce modèle, les clés primaires sont en italique et les clés étrangères sont précédées du symbole #. Un projet est toujours piloté par un ingénieur en chef, lui-même enregistré dans la table **Ingenieurs**. Une tâche est réalisée par un ingénieur et relativement à un projet spécifique. La livraison d'un projet est faite par un fournisseur.

Ecrire les requêtes SQL permettant de répondre aux questions suivantes.

- Q19.** Afficher le nom des ingénieurs et le nombre total de tâches qu'ils ont accomplies dans des projets avec un budget > 500 000 FCFA.
- Q20.** Afficher les tâches (description et durée) dont la durée dépasse la durée moyenne des tâches de leur projet.
- Q21.** Afficher les noms de fournisseurs ayant livré pour des projets dirigés par un ingénieur en "Mécanique".
- Q22.** Afficher les noms de fournisseurs n'ayant jamais livré pour des projets dirigés par un ingénieur en "Mécanique"
- Q23.** Afficher les pays où le coût total des livraisons dépasse 1 000 000 FCFA.
- Q24.** Afficher les projets en cours (date de fin non dépassée), triés par budget décroissant.
- NB : On suppose que la date du jour est donnée par la fonction SQL : **CURRENT_DATE ()**
- Q25.** Afficher le nom des ingénieurs ayant travaillé sur des projets de Mécanique et Électronique.