

BACCALAURÉAT GÉNÉRAL

CORRECTION DE L'ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

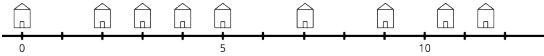
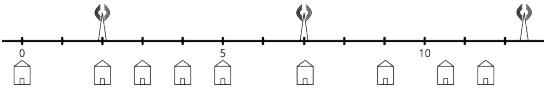
SESSION 2024

NUMÉRIQUE ET SCIENCES INFORMATIQUES

Durée de l'épreuve : 3 heures 30

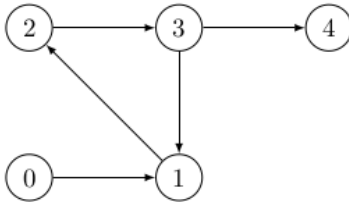
Le sujet est composé de trois exercices indépendants.

Exercice 1 6 points

Questions	Contenu et notions	Capacités exigibles	Éléments de réponses et commentaires
1	Prog Python, instancier un objet	N1	<pre>m1 = Maison(1) m2 = Maison(3.5)</pre>
2	Prog Python, instancier un objet	N1	<pre>a = Antenne(2.5, 1)</pre>
3	Algorithmique, représenter graphiquement un pb	N1	
4	Prog Python, construire une liste en itérant	N2	<pre>def creation_rue(pos): pos.sort() maisons = [] for p in pos: m = Maison(p) maisons.append(m) return maisons</pre>
5	Prog Python, POO, ajouter une méthode à une classe	N2	Une solution possible est la suivante. <pre>def couvre(self, maison): # Code à compléter (plusieurs lignes) distance = abs(self.get_pos_antenne() - maison.get_pos_maison()) return distance <= self.get_rayon()</pre>
6	Algorithmique, comprendre et exécuter du code Python à la main	N2	[0, 3, 7, 10.5]
7	Algorithmique, comprendre et exécuter un algorithme à la main	N2	

Exercice 1		6 points	
8	Algorithmique, Prog Python, implémenter un algorithme (guidé)	N2	<pre> def strategie_2(maisons, rayon): ''' Prend en paramètre une liste de maisons et le rayon d'action des antennes et renvoie une liste d'antennes ''' # Comme dans strategie_1 on suppose que c'est déjà trié. # Mais on pourrait trier ainsi : # maisons.sort(key=lambda x: x.pos()) antennes = [Antenne(maisons[0].pos_maison() + rayon, rayon)] for m in maisons[1:]: if not antennes[-1].couvre(m): nouvelle_antenne = Antenne(m.pos_maison() + rayon, rayon) antennes.append(nouvelle_antenne) return antennes </pre>
9	Algorithmique, calculer la complexité temporelle d'une fonction	N2	Toutes les opérations dans les 2 fonctions se font en un coût qui ne dépend pas du nombre de maisons ou du nombre d'antennes, excepté la boucle. La méthode couvre est un simple calcul élémentaire. La méthode append dans une liste Python s'exécute en un temps indépendant de la taille de la liste. Dans les 2 cas, le corps de la boucle est exécuté $n - 1$ fois s'il y a n maisons. Le coût est donc linéaire en le nombre de maisons dans les 2 cas.

Exercice 2 6 points

Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1.	Vocabulaire des graphes.	Reconnaître une représentation. Niveau 1	Il s'agit d'un graphe orienté.
2.	Modéliser une situation sous la forme d'un graphe.	Comprendre une représentation. Niveau 2	Réaliser la tâche (f) puis la tâche (g) est possible. Réaliser la tâche (g) puis la tâche (f) n'est pas possible. Réaliser la tâche (i) puis la tâche (j) est possible (pas de dépendance entre ces tâches) Réaliser la tâche (j) puis la tâche (i) est possible (pas de dépendance entre ces tâches).
3.	Modéliser une situation sous la forme d'un graphe.	Comprendre une représentation. Niveau 2	Pour réaliser la tâche (k) il faut avoir réalisé les tâches (a), (i), (c), (h), (j). Enlever des points si autres tâches citées (même si elles sont indépendantes). L'ordre dans lesquelles les tâches sont citées n'est pas important.
4.	Vocabulaire des graphes	Repérer la présence d'un cycle dans un graphe. Niveau 1	Le graphe de la figure 1 ne contient pas de cycle (sinon on ne pourrait pas réaliser la recette sans violer une dépendance !). Lorsqu'on suit les arcs dans le sens des flèches, on ne revient jamais sur une tâche déjà réalisée.
5.	Modéliser une situation sous la forme d'un graphe.	Utiliser une représentation. Niveau 2	Plusieurs possibilités, dont 0 -> 2 -> 1 -> 3 -> 5 -> 4 ou 3 -> 5 -> 4 -> 2 -> 0 -> 1.
6.	Abstraction.	Modéliser une situation sous la forme d'un graphe. Mobiliser les concepts utiles. Niveau 2.	Quelque chose comme ça devrait être proposé.  Graphe question 6.
7.	Graphe, cycle.	Modéliser une situation sous la forme d'un graphe. Mobiliser les concepts utiles. Niveau 2.	Ce n'est pas possible, il y a un cycle entre les sommets 1,2,3. Toute justification de cette dépendance cyclique pourra être acceptée.

Exercice 2		6 points	
8.	Fonction récursive / exploration de graphe et détection de cycle.	Analyser le fonctionnement d'un programme récursif / Niveau 3	Les tableaux ouverts et fermes ne contiennent initialement que des valeurs False , autant que de sommets dans le graphe. Le résultat de l'appel mystere(M, 1, ouverts, fermes, resultat) est False . En effet, lorsque la fonction est exécutée avec le sommet 1 alors, puisque ouverts[1] et fermes[1] valent False , ouverts[1] prend la valeur True puis la fonction mystere est appelée récursivement avec le successeur de 1 qui est 2. A l'entrée dans la fonction, ouverts[2] et fermes[2] valent False , ouverts[2] prend la valeur True . La fonction est appelée récursivement avec le sommet 3 qui est le successeur de 2. À l'entrée dans la fonction, ouverts[3] et fermes[3] valent False , ouverts[3] prend la valeur True . La fonction est appelée récursivement avec le sommet 1 qui a déjà été visité : ouverts[1] vaut alors True et la fonction renvoie False , valeur renvoyée par tous les appels récursifs imbriqués.
9.	Fonction récursive, détection de graphe.	Compréhension de fonction, capacité de généralisation. / Niveau 3	La fonction renvoie False lorsqu'il y a un cycle dans le graphe ou toute autre explication en lien avec la liste des tâches.
10.	Langages et programmation, structure de Pile (abstraite), programmation orientée objet.	Compréhension de code / Niveau 1	La valeur de elt après l'exécution de la suite d'instructions est 3. Objectif : prise en main de la classe Pile proposée.
11.	Parcours en profondeur, utilisation d'une pile (pour le résultat !).	Initiative / Niveau 2	Ne rien mettre à la ligne 17. Remplacer la ligne 25 par resultat.empiler(s) . Lorsque tous les successeurs d'un sommet ont été traités, il est possible d'ajouter ce sommet au sommet de la pile. Résoudre cette question nécessite d'avoir bien compris l'ensemble du sujet.

Exercice 3		8 points	
Questions	Contenu et notions	Capacités exigibles / Niveau	Éléments de réponses et commentaires
1	maîtrise de la POO	N1	<code>personneA = Personne(112, 'LESIEUR', 'Isabelle', 1982, 2005)</code>
2	maîtrise de la POO	N1	<code>personneA.num_badge</code>
3	écriture d'une classe en Python	N2	<code>def annee_anciennete(self): return 2024 - self.annee_entree</code>
4	programmation Python, POO	N2	<code>def ajouter(self, personne): self.liste.append(personne)</code>
5	programmation Python, POO	N2	<code>def effectif(self): return len(self.liste)</code>
6	programmation Python, POO	N2	<code>def donne_nom(self, num): for elt in self.liste: if elt.num_badge == num : return elt.nom return None</code>
7	programmation Python, POO	N3	<code>def nb_personne_honneur(self, annee): nb = 0 for elt in self.liste: if annee - elt.annee_entree == 10 : nb = nb + 1 return nb def nb_personne_honneur(self): nb = 0 for elt in self.liste: if elt.annee_anciennete() == 10 : nb = nb + 1 return nb</code> ou un autre code qui fonctionne

Exercice 3		8 points	
8	programmation Python, POO	N3	<pre>def plus_anciens(self): annee_min = self.liste[0].annee_entree liste = [] for elt in self.liste: if elt.annee_entree < annee_min : annee_min = elt.annee_entree liste = [elt.num_badge] elif elt.annee_entree == annee_min : liste.append(elt.num_badge) return liste</pre> ou un autre code qui fonctionne
9	base de données, requête	N1	retourne les noms et prénoms des personnes travaillant dans le centre numéro 2.
10	base de données, requête de mise à jour	N2	<pre>UPDATE Personnel SET num_centre = 3 WHERE nom = 'HADJI' AND prenom = 'Hakim';</pre> ou sql <code>UPDATE Personnel SET num_centre = 3 WHERE num_badge = 135;</code>
11	base de données	N1	éviter la duplication de données ou il est préférable que la table contienne des informations relatives au meme sujet.
12	base de données	N1	à l'aide d'une clé étrangère qui fait référence à une clé primaire <code>num</code> de la table <code>centre</code>
13	base de données, requête avec jointure	N3	<pre>SELECT Personnel.nom FROM Personnel JOIN centre ON Personnel.num_centre = centre.num WHERE centre.ville = 'Lille' AND 2015<=Personnel.annee_debut<=2020;</pre>
14	base de données	N3	car dans ce cas les gens qui travaillaient à Caen auront une clé étrangère <code>num_centre</code> qui ne fera référence à aucune valeur de l'attribut <code>num</code> de la table <code>Centre</code> .