

**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		
<i>Questions</i>	<i>Contenu et notions</i>	<i>Capacités exigibles / Niveau</i>	<i>Barème</i>	<i>Éléments de réponses et commentaires</i>
1	Appliquer une fonction donnée sur un exemple donné	N1		La variable <i>i</i> prend les valeurs 7,8,9,10 ; puis <i>f1(7)</i> termine en revoyant 10.
2	Appliquer une fonction donnée sur un exemple donné	N1		Tout comme dans la question précédente, la variable <i>i</i> va être incrémentée jusqu'à valoir 10 : ainsi, <i>f1(-2)</i> termine et renvoie 10.
3	Appliquer une fonction donnée sur un exemple donné	N1		La variable <i>i</i> va successivement valoir 12, 13, 14, 15, 16, ... et ne vaudra jamais 10 : ainsi, <i>f1(12)</i> ne termine pas.
4	Comprendre le comportement d'une fonction donnée	N2		<i>f1(n)</i> termine sur tous les entiers $n \leq 10$ (et renvoie 10 dans ce cas).
5	Appliquer une fonction récursive donnée sur un exemple donné	N1		<i>f2(4)</i> calcule $4 + f2(2)$ qui calcule alors $4 + 2 + f2(0)$ . Et <i>f2(0)</i> termine et renvoie 0 : ainsi, <i>f2(4)</i> termine et renvoie $4 + 2 + 0 = 6$ .
6	Appliquer une fonction donnée sur un exemple donné	N1		<i>f2(5)</i> va faire un appel récursif sur <i>f2(3)</i> puis <i>f2(1)</i> puis <i>f2(-1)</i> ... Ainsi, on ne va jamais passer dans le cas de base $n==0$ : <i>f2(5)</i> ne va jamais terminer.
7	Comprendre le comportement d'une fonction récursive donnée	N1		<i>f2(n)</i> termine sur tous les entiers naturels pairs (et s'exécute à l'infini sur tous les entiers impairs).

---

Exercice 1	6 points		
8	Écrire une fonction Python respectant une spécification donnée	N2	<p>On souhaite une fonction impérative ayant une boucle infinie ou une fonction récursive qui effectue des appels récursifs à l'infini.</p> <pre>def infini(n):     return infini(n)</pre>
9	Comprendre le comportement d'une fonction sur un cas particulier	N2	<p>Si <code>arret(code_paradoxe, code_paradoxe)</code> renvoie <code>True</code>, la prochaine instruction à être exécutée est <code>infini(42)</code>, qui ne va pas terminer.</p>
10	Comprendre le comportement d'une fonction sur un cas particulier	N2	<p>Si <code>arret(code_paradoxe, code_paradoxe)</code> renvoie <code>False</code>, la prochaine instruction à être exécutée est <code>return 0</code>, la fonction termine donc instantanément.</p>
11	Expliquer qu'un problème est indécidable.	N3	<p>Les questions précédentes nous montrent que l'existence d'une telle fonction <code>arret</code> est paradoxal : on arriverait alors à écrire une fonction <code>paradoxe</code> qui termine si et seulement si elle ne termine pas... Il est donc impossible d'écrire une telle fonction <code>arret</code>.</p>

---

---

**Exercice 2**    6 points

---

<i>Questions</i>	<i>Contenu et notions</i>	<i>Capacités exigibles / Niveau</i>	<i>Barème</i>	<i>Éléments de réponses et commentaires</i>
1	Appliquer à la main un algorithme simple utilisant des piles	N1		Étape 1 : 3->2->1, 4->5->6 vide Étape 2 : 3->2->1, vide 6->5->4 Étape 3 : vide, 4->1->5->2->6->3, vide
2	Compléter un code Python simple	N1		<pre>for i in range(n, 0, -1):     resultat.empile(i) ou bien for i in range(n):     resultat.empile(n - i)</pre>
3	Analyser et rectifier un programme Python	N1		Dans les lignes 4 et 5, remplacer n par n // 2 et dans la ligne 3 ajouter les parenthèses m2 = Pile()
4	Traduire un algorithme en programme Python	N2		<pre>def recombinaer(m1, m2):     p = Pile()     while not m1.est_vide():         p.empile(m1.depile())         p.empile(m2.depile())     return p</pre>
5	Mobiliser différentes fonctions Python pour produire une tâche plus complexe	N2		<pre>def faro(p, n):     m1, m2 = scinder_jeu(p, n)     return recombinaer(m1, m2)</pre>
6	Produire un jeu de tests	N1		On ne pénalise pas l'absence des "tests symétriques" <pre>p2.empile(1) assert identiques(p1, p2) # piles non vides identiques p2.empile(2) p1.empile(3) assert identiques(p1,p2) == False # même taille mais différentes</pre>

---

Exercice 2 6 points

---

7 Résoudre un problème N3  
en autonomie et le  
programmer en  
Python

Toute trace de recherche sera valorisée. L'utilisation d'une fonction de copie de pile sera tolérée et amènera à une note maximale de 1,5 pt

```
def ordre_faro(n):  
    jeu = produire_jeu(n)  
    jeu_original = produire_jeu(n)  
    jeu = faro(jeu, n)  
    repetitions = 1  
    while not identiques(jeu, jeu_original):  
        jeu = faro(jeu, n)  
        repetitions += 1  
    return repetitions
```

---

Exercice 3 8 points

Questions	Contenu et notions	Capacités exigibles / Niveau	Barème	Éléments de réponses et commentaires
1	Réseau; adressage IP	N1		Une adresse IPv4 est composée de 4 octets
2	Réseau; adressage IP	N1		<ul style="list-style-type: none"> <li>• Serveur_web : 172.16.0.1</li> <li>• Serveur_BDD : 172.16.0.2</li> </ul>
3	Réseau	N1		Tester l'accessibilité d'une autre machine à travers un réseau IP
4	Réseau; adressage IP	N1		L'adresse de la passerelle est erronée / corriger l'adresse de la passerelle sur PC_A1 en 192.168.1.254
5	Routage	N1		PC_A1 -> A -> B -> C -> D -> Serveur_impression
6	Routage	N1		PC_A1 -> A -> B -> C X / Les paquets n'arrivent pas à destination
7	Routage	N1		<ul style="list-style-type: none"> <li>• 192.168.1.0   <b>10.0.2.2</b>   <b>2</b></li> <li>• 192.168.2.0   <b>10.0.3.2</b>   <b>1</b></li> <li>• 192.168.3.0   <b>10.0.4.2</b>   <b>1</b></li> <li>• 10.0.1.0   <b>10.0.2.2</b>   <b>1</b></li> <li>• 10.0.5.0   <b>10.0.3.2</b>   <b>1</b></li> <li>• ou 10.0.5.0   <b>10.0.4.2</b>   <b>1</b></li> </ul>
8	Routage	N1		PC_A1 -> A -> B -> C -> D -> Serveur_impression
9	algorithme sur les graphes; DFS	N1		Entre les routeurs C et D, le débit est de 10Mb/s alors qu'en passant par E le débit est de 1Gb/s
10	Routage	N2		dest : 192.168.2.0 en passant par 10.0.4.2 avec 2 sauts
11	Bases de données SQL	N1		<b>SELECT titre_parution FROM parution;</b>
12	Bases de données SQL	N1		les numéros de parutions triés dans l'ordre croissant et les numero de pages de la table <b>page</b> pour lesquels la mise en forme est 'Arial,12'.
13	Bases de données SQL	N1		<b>SELECT num_image, titre_image, poids FROM image WHERE poids &gt; 1000;</b>
14	Bases de données SQL	N1		La liste des numéros de parutions où apparaissent des images dont le titre contient le mot "Apollo".

15	Bases de données SQL	N1	Ajoute une nouvelle image à la table <code>image</code> avec les caractéristiques suivantes : <ul style="list-style-type: none"><li>• <code>num_image = 2923</code>,</li><li>• <code>titre_image = 'Volcans du massif central'</code>,</li><li>• <code>descriptif = ''</code>,</li><li>• <code>largeur = 400</code>,</li><li>• <code>hauteur = 400</code>,</li><li>• <code>poids = 1430</code></li></ul>
16	Bases de données	N1	<pre>INSERT INTO texte VALUES(2754, 'Vulcania', "Parc d'attraction", 250);</pre>
17	Bases de données	N1	Supprime l'enregistrement de la table <code>texte</code>
18	Bases de données	N1	<pre>DELETE FROM comporte_texte WHERE num_texte = 2034;</pre>

---