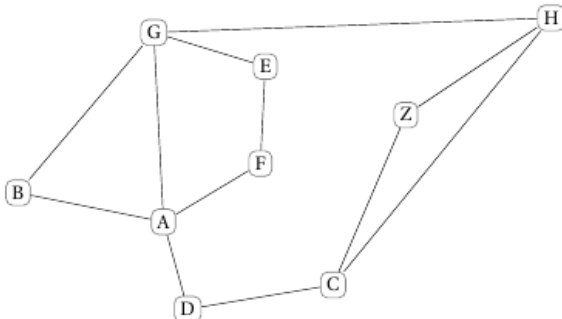


## **NUMERIQUE ET SCIENCES INFORMATIQUES – CORRIGÉ – JOUR 2**

Recommandations :

- Pour les parties de code rédigées en Python ou en SQL, les erreurs de syntaxe élémentaires (oubli des guillemets, des « : », etc.) ne seront pas pénalisées si, sur le fond, la question est correctement traitée.
- Une partie d'une question ou d'un code traitée correctement sera valorisée.

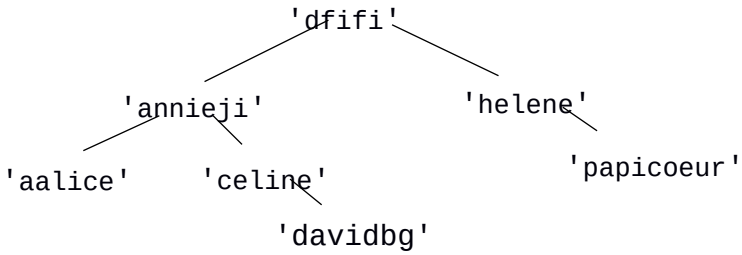
### EXERCICE 1 : PROTOCOLES RESEAUX (3 points)

EXERCICE 1 : PROTOCOLES RESEAUX (3 points)																																								
	Contenus et notions	Capacités exigibles/Niveau	Éléments de réponses et commentaires	Barème																																				
1.	Réseau	Appliquer un protocole.	<table><thead><tr><th colspan="2">Nœud B</th></tr><tr><th>Destination</th><th>Coût</th></tr></thead><tbody><tr><td>A</td><td>1</td></tr><tr><td>C</td><td>3</td></tr><tr><td>D</td><td>2</td></tr><tr><td>E</td><td>2</td></tr><tr><td>F</td><td>2</td></tr><tr><td>G</td><td>1</td></tr><tr><td>H</td><td>2</td></tr></tbody></table> <table><thead><tr><th colspan="2">Nœud F</th></tr><tr><th>Destination</th><th>Coût</th></tr></thead><tbody><tr><td>A</td><td>1</td></tr><tr><td>B</td><td>2</td></tr><tr><td>C</td><td>3</td></tr><tr><td>D</td><td>2</td></tr><tr><td>E</td><td>1</td></tr><tr><td>G</td><td>2</td></tr><tr><td>H</td><td>3</td></tr></tbody></table>	Nœud B		Destination	Coût	A	1	C	3	D	2	E	2	F	2	G	1	H	2	Nœud F		Destination	Coût	A	1	B	2	C	3	D	2	E	1	G	2	H	3	0,5
Nœud B																																								
Destination	Coût																																							
A	1																																							
C	3																																							
D	2																																							
E	2																																							
F	2																																							
G	1																																							
H	2																																							
Nœud F																																								
Destination	Coût																																							
A	1																																							
B	2																																							
C	3																																							
D	2																																							
E	1																																							
G	2																																							
H	3																																							
2.	Réseau	Compréhension d'un protocole.	F-E-G-H et F-A-G-H <b>0,5 par chemin correct</b> <b>0 dès que d'autres chemins sont proposés.</b>	1																																				
3.	Réseau	Compréhension d'un protocole.	 <p>Il est aussi possible de placer le nœud Z sur l'arête [CH]</p>	0,75																																				

<b>4.</b>	Réseau	Appliquer un protocole.	<p>B-G-E-F-A-D-C-H</p> <p>Justification : Préciser que ce parcours possède le coût le plus petit (34).</p> <p>L'élève peut aussi lister l'ensemble des chemins possibles avec leur coût.</p>	<b>0,75</b>
-----------	--------	-------------------------	--	-------------

EXERCICE 2 : Base de données et langage SQL (3 points)				
	Contenus et notions	Capacités exigibles/Niveau	Éléments de réponses et commentaires	Barème
<b>1.</b>				
<b>a.</b>	BDD	Connaitre la notion de clé primaire.	Une clé primaire est un attribut permettant d'identifier de manière unique un enregistrement dans une table.	<b>0,25</b>
<b>b.</b>	BDD	Connaitre les contraintes liées à la clé primaire.	Cette requête ajoute un nouvel enregistrement dans la table Astronaute dont la clé primaire est déjà existante.	<b>0,25</b>
<b>c.</b>	BDD	Écrire le schéma relationnel d'une table.	<p>Fusee(<u>id_fusee</u> : INTEGER, modele : TEXT, constructeur : TEXT, nb_places : INTEGER)</p> <p>Pour le type chaîne de caractères, on acceptera CHAINE DE CARACTERES, TEXT, STR, CHAR ...</p> <p>Pour le type entier, on acceptera ENTIER, INT, INTEGER.</p> <p>Toute autre représentation correcte sera validée.</p> <p>L'identification de la clé primaire n'est pas évaluée.</p>	<b>0,25</b>
<b>2.</b>				
<b>a.</b>	BDD	Connaitre le langage SQL.	2	<b>0,25</b>
<b>b.</b>	BDD	Écrire une requête SQL.	<p>SELECT modele, constructeur FROM Fusee WHERE nb_places &gt;= 4;</p> <p>ou</p> <p>SELECT modele, constructeur FROM Fusee WHERE nb_places &gt; 3;</p>	<b>0,5</b>

<b>c.</b>	BDD	Écrire une requête SQL.	SELECT nom, prenom FROM Astronaute ORDER BY nom;	<b>0,5</b>
<b>3.</b>				
<b>a.</b>	BDD	Écrire une requête SQL Repérer les clés étrangères dans une table.	INSERT INTO Vol VALUES (5, 3, '12/04/2023'); INSERT INTO Equipe VALUES(5, 1); INSERT INTO Equipe VALUES(5, 4);	<b>0,75</b>
<b>b.</b>	BDD	Écrire une requête SQL Connaitre la notion de jointure entre 2 tables.	SELECT nom, prenom FROM Astronaute JOIN Equipe ON Astronaute.id_astronaute = Equipe.id_astronaute JOIN Vol ON Equipe.id_vol = Vol.id_vol WHERE Vol.date ='25/10/2022'  <i>0,25 pour le WHERE. Le JOIN n'est pas évalué.</i>	<b>0,25</b>

EXERCICE 3 : (6 points)				
	Contenus et notions	Capacités exigibles/Niveau	Eléments de réponses et commentaires	Barème
1.	Arbre binaire	Évaluer la taille et hauteur d'un arbre.	taille = 5 hauteur = 3  <i>0,25 par réponse correcte.</i>	0,5
2.	Arbre binaire de recherche	Insérer des éléments dans un ABR.	 <pre> graph TD     dfifi["'dfifi'"] --&gt; annieji["'annieji'"]     dfifi --&gt; helene["'helene'"]     annieji --&gt; aalice["'aalice'"]     annieji --&gt; celine["'celine'"]     celine --&gt; davidbg["'davidbg'"]     helene --&gt; papicoeur["'papicoeur'"] </pre> <i>0,25 pour 'papicoeur' et 0,5 pour 'davidbg'.</i>	0,75
3.	Parcours d'un arbre	Connaitre les différents parcours d'un arbre.	C. Parcours dans l'ordre infixe	0,5

4.	POO	Connaitre l'algorithme pour insérer un élément dans un ABR arbre et maîtrise de la POO.	<pre>def present(self, identifiant) :     if self.est_vide() :         return False     elif self.racine() == identifiant :         return <b>True</b>     elif self.racine() &lt; identifiant :         return self.sd().<b>present(identifiant)</b>     else :         return <b>self.sg().present(identifiant)</b></pre> <p><b>0,25 pour True</b></p>	0,5					
5.a	File	Comprendre la notion de file.	False	0,25					
5.b	File	Comprendre la notion de file.	<table border="1"> <tr> <td></td> <td>'bac'</td> <td>'nsi'</td> <td>'2023'</td> <td></td> </tr> </table>		'bac'	'nsi'	'2023'		0,25
	'bac'	'nsi'	'2023'						
5.c	File	Comprendre la notion de file.	<table border="1"> <tr> <td></td> <td>'poule'</td> <td>'python'</td> <td>'castor'</td> <td></td> </tr> </table>		'poule'	'python'	'castor'		0,25
	'poule'	'python'	'castor'						
6.	File et programmation	Maitriser les files et la programmation de base.	<pre>def longueur(f) :     resultat = 0     g = creer_file()     while <b>not(est_vide(f))</b> :         elt = defiler(f)         resultat = <b>resultat + 1</b>         <b>enfiler(g , elt)</b>     while not(est_vide(g)):         enfiler(f, defiler(g))     return resultat</pre> <p><b>0,25 par réponse correcte.</b></p>	0,75					

7.	Programmation de base et booléen	Comprendre un code.	Réponse C ou '2!@59fgds'	0,25
8.	File	Utiliser les fonctions sur les files.	<pre>def ajouter_mot(f, mdp):     enfiler(f, mdp)     if longueur(f) &gt; 3:         defiler(f)</pre> Ou tout autre code valide.	0,75
9.	File	Utiliser les fonctions sur les files.	<pre>def mot_file(f,mdp) :     g = creer_file()     present = False     while not(est_vide(f)):         elt = defiler(f)         enfiler(g , elt)         if elt == mdp :             present = True     while not(est_vide(g)) :         enfiler(f, defiler(g))     return present</pre> Les deux réponses doivent être correctes.	0,25
10.	Programmation	Écrire une fonction.	<pre>def modification(f, nv_mdp) :     if mot_file(f, nv_mdp) or not(est_valide(nv_mdp)):         return False     else :         ajouter_mot(f, nv_mdp)         return True</pre> ou <pre>def modification(f, nv_mdp) :     if not(est_valide(nv_mdp)) :         return False     if mot_file(f, nv_mdp) :         return False</pre>	1



			<pre> else:     ajouter_mot(f, nv_mdp)     return True ou def modification(f, nv_mdp) :     if est_valide(nv_mdp) and not(mot_file(f, nv_mdp)) :         ajouter_mot(f, nv_mdp)         return True     else:         return False ou tout autre code valide </pre>	
--	--	--	---	--