

*Exercice 1 (6 points)*

**Correction**

Question	Barème	Niveau	Contenu	Solution												
1				Routeur R2 via son interface G0.												
2				L'adresse de diffusion est 172.16.255.255 donc celle du routeur est 172.16.255.254.												
3				Il y a 4 adresses 50.50.50.4 (adresse réseau), 50.50.50.5 (interface G2), 50.50.50.6 et 50.50.50.7 (adresse de diffusion). Ainsi 50.50.50.6 est l'adresse du routeur R4.												
4				<p>Table de routage du routeur R1</p> <hr/> <table><thead><tr><th>Réseau de destination</th><th>Interface de sortie</th><th>Métrique</th></tr></thead><tbody><tr><td>192.168.1.0/24</td><td>G0</td><td>0</td></tr><tr><td>192.168.2.0/24</td><td>G3</td><td>1</td></tr><tr><td>192.168.3.0/24</td><td>G3</td><td>2</td></tr></tbody></table>	Réseau de destination	Interface de sortie	Métrique	192.168.1.0/24	G0	0	192.168.2.0/24	G3	1	192.168.3.0/24	G3	2
Réseau de destination	Interface de sortie	Métrique														
192.168.1.0/24	G0	0														
192.168.2.0/24	G3	1														
192.168.3.0/24	G3	2														

Question	Barème	Niveau	Contenu	Solution		
				172.16.0.0/16 10.0.0.0/8 50.50.50.0/30 50.50.50.4/30 50.50.50.8/30 50.50.50.12/30 50.50.50.16/30 50.50.50.20/30 50.50.50.24/30	G2 G1 G1 G2 G3 G2 G3 G1 G2	2 1 0 0 0 1 1 1 1
5				<pre>1 t_routage = [ ((192, 168, 1, 0, 24), 'G0', 0), 2                 ((192, 168, 2, 0, 24), 'G3', 1), 3                 ((192, 168, 3, 0, 24), 'G3', 2), 4                 ((172, 16, 0, 0, 16), 'G2', 2), 5                 ((10, 0, 0, 0, 8), 'G1', 1), 6                 ((50, 50, 50, 0, 30), 'G1', 0), 7                 ((50, 50, 50, 4, 30), 'G2', 0), 8                 ((50, 50, 50, 8, 30), 'G3', 0), 9                 ((50, 50, 50, 12, 30), 'G2', 1), 10                ((50, 50, 50, 16, 30), 'G3', 1), 11                ((50, 50, 50, 20, 30), 'G1', 1), 12                ((50, 50, 50, 24, 30), 'G2', 1) 13 ]</pre>		

Question	Barème	Niveau	Contenu	Solution								
6				<pre>&gt;&gt;&gt; mask_for_size(30) (255, 255, 255, 252)</pre>								
7				$50 = 32 + 16 + 2 = (0011\ 0010)_2$ <table style="margin-left: 200px; border-collapse: collapse;"> <tr> <td style="border-right: 1px dashed black;">adresse IP :</td> <td>00110010.00110010.00110010.00000110</td> </tr> <tr> <td style="border-right: 1px dashed black;">masque :</td> <td>11111111.11111111.11111111.11111100</td> </tr> <tr> <td style="border-right: 1px dashed black;">&amp;</td> <td>-----</td> </tr> <tr> <td></td> <td>00110010.00110010.00110010.00000100</td> </tr> </table> <p>L'adresse IP 50.50.50.6/30 appartient au réseau 50.50.50.4/30.</p>	adresse IP :	00110010.00110010.00110010.00000110	masque :	11111111.11111111.11111111.11111100	&	-----		00110010.00110010.00110010.00000100
adresse IP :	00110010.00110010.00110010.00000110											
masque :	11111111.11111111.11111111.11111100											
&	-----											
	00110010.00110010.00110010.00000100											
8				<pre>def is_in_network(address, network):     network_mask = mask_for_size(network[4])     for i in range(4):         if et_bit_a_bit(network_mask[i], address[i]) != network[i]:             return False     return True</pre>								
9				<pre>def choose_interface(t_routage, destination_ip):     for i in range(len(t_routage)):         network, interface, metrique = t_routage[i]         if is_in_network(destination_ip, network):</pre>								

Question	Barème	Niveau	Contenu	Solution																											
				<pre><code>return interface return None # rq : le return est facultatif</code></pre> <p>ou toute autre implémentation correcte</p>																											
10				<table> <thead> <tr> <th>Type</th> <th>Débit</th> <th>Coût</th> </tr> </thead> <tbody> <tr> <td>Fast Ethernet</td> <td>100 Mb/s</td> <td>100</td> </tr> <tr> <td>Gigabit Ethernet</td> <td>1 Gb/s</td> <td>10</td> </tr> <tr> <td>Fibre</td> <td>10 Gb/s</td> <td>1</td> </tr> </tbody> </table>	Type	Débit	Coût	Fast Ethernet	100 Mb/s	100	Gigabit Ethernet	1 Gb/s	10	Fibre	10 Gb/s	1															
Type	Débit	Coût																													
Fast Ethernet	100 Mb/s	100																													
Gigabit Ethernet	1 Gb/s	10																													
Fibre	10 Gb/s	1																													
11				<p>Table de routage du routeur R1</p> <table> <thead> <tr> <th>Réseau de destination</th> <th>Interface de sortie</th> <th>Métrique</th> </tr> </thead> <tbody> <tr> <td>192.168.1.0/24</td> <td>G0</td> <td>0</td> </tr> <tr> <td>192.168.2.0/24</td> <td>G3</td> <td>10</td> </tr> <tr> <td>192.168.3.0/24</td> <td>G3</td> <td>20</td> </tr> <tr> <td>172.16.0.0/16</td> <td>G3</td> <td>12</td> </tr> <tr> <td>10.0.0.0/8</td> <td>G1</td> <td>10</td> </tr> <tr> <td>50.50.50.0/30</td> <td>G1</td> <td>0</td> </tr> <tr> <td>50.50.50.4/30</td> <td>G2</td> <td>0</td> </tr> <tr> <td>50.50.50.8/30</td> <td>G3</td> <td>0</td> </tr> </tbody> </table>	Réseau de destination	Interface de sortie	Métrique	192.168.1.0/24	G0	0	192.168.2.0/24	G3	10	192.168.3.0/24	G3	20	172.16.0.0/16	G3	12	10.0.0.0/8	G1	10	50.50.50.0/30	G1	0	50.50.50.4/30	G2	0	50.50.50.8/30	G3	0
Réseau de destination	Interface de sortie	Métrique																													
192.168.1.0/24	G0	0																													
192.168.2.0/24	G3	10																													
192.168.3.0/24	G3	20																													
172.16.0.0/16	G3	12																													
10.0.0.0/8	G1	10																													
50.50.50.0/30	G1	0																													
50.50.50.4/30	G2	0																													
50.50.50.8/30	G3	0																													

<i>Question</i>	<i>Barème</i>	<i>Niveau</i>	<i>Contenu</i>	<i>Solution</i>		
				50.50.50.12/30	G3	10
				50.50.50.16/30	G3	10
				50.50.50.20/30	G1	10
				50.50.50.24/30	G3	11

*Exercice 2 (6 points)*

**Correction**

Question	Barème	Niveau	Contenu	Solution
1		1	comprendre la définition du problème	La taille est 3 La base est issue de la cellule <code>carte_B[0][1]</code>
2		1	utiliser une modélisation	Si ce carré est constructible, il ne contient que des valeurs 1. La somme de ces valeurs est égale au produit de sa largeur par sa hauteur et vaut donc <code>taille * taille</code> .
3		2	compléter une fonction Python simple	<pre><code>def est_constructible(carte, i_coin, j_coin, taille):     s = 0     for i in range(i_coin, i_coin + taille):         for j in range(j_coin, j_coin + taille):             s += carte[i][j]     return s == taille * taille</code></pre>
4		3	compléter une fonction Python complexe	<pre><code>def plus_grande_base_exhaustive(carte):     n = len(carte)     # les tailles vont de n à 1, de -1 en -1     for taille in range(n, 0, -1):         i = 0         while i + taille &lt;= n:             j = 0             while j + taille &lt;= n:                 if est_constructible(carte, i, j,</code></pre>

Question	Barème	Niveau	Contenu	Solution
				<pre> taille):     return (taille, i, j)     j = j + 1     i = i + 1 </pre>
5		1	utiliser la notion de coût d'un algorithme	<p>Le nombre de carré à tester devient rapidement très grand. Dans le cas de grandes cartes, le temps de calcul est si important qu'une telle recherche exhaustive devient impraticable.</p> <p>On attend que le candidat formule une réponse qui tient compte du temps mis par le calcul.</p> <p>Pour information, le nombre de carrés à tester est égal à <math>\frac{n(n+1)(2n+1)}{6}</math>.</p>
6		1	utiliser une définition	<pre> aux_B = [     [1, 3, 2, 1],     [0, 2, 2, 1],     [0, 1, 2, 1],     [1, 0, 1, 1] ] </pre>
7		2	raisonner	<p>Comme <code>carte_C[0][3] = 0</code> cette cellule n'est pas constructible. On a donc <code>a = aux_C[0][3] = 0</code>.</p>

Question	Barème	Niveau	Contenu	Solution
				<p>Comme <code>carte_C[1][0] = 1</code>, il est possible de construire une base carré de taille 1 issue de cette cellule.</p> <p>On lit par contre <code>aux_C[1][1] = 0</code>. Il n'existe donc pas de base carrée issue de la cellule <code>carte_C[1][1]</code> ce qui signifie que <code>carte_C[1][1] = 0</code>. Il est donc impossible de construire une base de taille supérieure ou égale à 2 issue de <code>tab[1][1]</code>.</p> <p>On a donc <code>b = aux_C[1][0] = 1</code>.</p> <p>Des raisonnements analogues montrent que <code>c = aux_C[3][3] = 3</code> et que <code>d = aux_C[4][0] = 2</code></p>
8		2	compléter une fonction Python	<pre> <b>def</b> calcule_aux(carte):     n = <b>len</b>(carte)     aux = [[0 <b>for</b> j <b>in</b> <b>range</b>(n)] <b>for</b> i <b>in</b> <b>range</b>(n)]      <i># Remplissage de la dernière ligne</i>     <i># et de la dernière colonne</i>     <b>for</b> k <b>in</b> <b>range</b>(n):         aux[n - 1][k] = carte[n - 1][k]         aux[k][n - 1] = carte[k][n - 1]      <i># On complète les lignes de bas en haut</i>     <b>for</b> i <b>in</b> <b>range</b>(n - 2, -1, -1):         <i># On complète les colonnes de droite à gauche</i>         <b>for</b> j <b>in</b> <b>range</b>(n - 2, -1, -1):             <b>if</b> carte[i][j] == 1:                 aux[i][j] = 1 + <b>min</b>(aux[i + 1][j],                 aux[i][j + 1],                 aux[i + 1][j + 1]) </pre>

Question	Barème	Niveau	Contenu	Solution
				<pre> aux[i + 1][j + 1]) return aux </pre>
9		2	compléter une fonction Python	<pre> def plus_grande_base(carte):     n = len(carte)     aux = calcule_aux(carte)     taille_max = aux[0][0]     i_max = 0     j_max = 0     for i in range(1, n):         for j in range(1, n):             if taille_max &lt; aux[i][j]:                 taille_max = aux[i][j]                 i_max = i                 j_max = j     return (taille_max, i_max, j_max) </pre>
10		2	estimer le coût d'exécution d'une fonction	<p>La fonction <code>plus_grande_base</code> complète la liste auxiliaire puis cherche la valeur maximale de celle-ci.</p> <p>Si la carte initiale est de dimension <math>n</math>, il faut donc calculer <math>n^2</math> valeurs pour compléter la liste auxiliaire et chercher le maximum parmi ces <math>n^2</math> valeurs.</p> <p>Donc, lorsque la largeur de la carte triple, le nombre de valeurs à calculer est multiplié par <math>3^2 = 9</math>.</p> <p>On peut donc estimer le temps d'exécution à <math>0,4 \times 9 = 3,6</math> secondes.</p>

*Exercice 3 (8 points)*

**Correction**

Question	Barème	Niveau	Contenu	Solution
1		1	Base de données : rôle d'un attribut	Une clé primaire est un attribut dont la valeur est unique dans la table.
2		1	Base de données : rôle d'un attribut	L'attribut habitat ne peut pas en être une car il existe plusieurs espèces de papillons qui ont le même habitat.
3		1	Base de données : requête SQL	85
4		2	Base de données : requête SQL	<code>UPDATE papillon SET taille = 50 WHERE numCo = 'Petite Tortue'</code>  On peut accepter num=462 même si l'énoncé ne précise pas que num est une clé primaire.
5		2	Base de données :	<code>SELECT nomCo FROM papillon WHERE habitat = 'Prairies' AND taille &lt; 55 ;</code>

Question	Barème	Niveau	Contenu	Solution
			requête SQL	
6		1	Base de données : requête SQL	Papaver rhoeas
7		2	Base de données : requête SQL	<pre><b>SELECT</b> papillon.nomCo, plante.nomCo <b>FROM</b> papillon <b>JOIN</b> plante <b>ON</b> plante.habitat = papillon.habitat <b>WHERE</b> taille &lt; 55 ;</pre>
8		1	Base de données : requête SQL	Paon-du-jour lilas : Le papillon et la plante qui sont tous deux en Europe.
9		2	Base de données : requête SQL	<pre><b>SELECT</b> papillon.nomCo <b>FROM</b> papillon <b>JOIN</b> zone_geographique <b>ON</b> papillon.zone = zone_geographique.num <b>JOIN</b> plante <b>ON</b> plante.zone = zone_geographique.num <b>WHERE</b> plante.nomCo='Coquelicot'</pre>
10		2	Langage et programm	<pre>1 <b>def</b> tri_collec(collec): 2     """Renvoie la collection des papillons triées 3     par ordre croissant de leur taille.</pre>

Question	Barème	Niveau	Contenu	Solution
			ation : compléter une fonction simple	<pre> 4     Paramètre: 5         collec : liste de dictionnaires des papillons 6     Renvoie: 7         liste triée par ordre croissant des tailles 8         des papillons. 9     """ 10    for i in range(1, len(collec)): 11        pap = collec[i] 12        j = i 13        while j &gt; 0 and collec[j - 1]['taille'] &gt; 14            pap['taille']: 15            collec[j] = collec[j - 1] 16            j = j - 1 17            collec[j] = pap 18    return collec </pre>
11		1	Algorithmique : tris	La fonction <code>tri_collec</code> réalise un tri par insertion.
12		2	Algorithmique : complexité	Le coût est quadratique, de l'ordre de $O(n^2)$ car il y a la présence de deux boucles imbriquées, où le corps est en temps constant, pour chaque valeur de <code>i</code> , la valeur de <code>j</code> est testée.
13		1	Algorithmique :	L'algorithme des $k$ plus proches voisins (KNN pour K Nearest Neighbors en anglais) est un algorithme d'apprentissage automatique (machine learning).

Question	Barème	Niveau	Contenu	Solution
				Un nouvel élément peut être catégorisé selon la classe majoritaire de ses $k$ plus proches voisins déterminés grâce à un grand jeu de données.
14		3	Algorithme : recherche textuelle	<pre> 1 def recherche_seq(seq, chaine): 2     """Renvoie l'indice du premier caractère de 3         chaine où commence `seq` si la séquence `seq` 4         se trouve dans la chaine de caractères chaine, 5         -1 sinon 6     Paramètres: 7         seq : séquence à rechercher 8         chaine : chaine d'ADN 9     Renvoie: 10        indice du premier caractère de seq dans 11        la chaine, -1 sinon. 12    """ 13    for i in range(len(chaine)-len(seq) + 1): 14        j = 0 15        while j &lt; len(seq) and chaine[i + j] == seq[j]: 16            j += 1 17        if j == len(seq): 18            return i 19    return -1 </pre>
15		2	Algorithme : recherche textuelle	Dans cet algorithme, la séquence <code>seq</code> est comparée à la chaîne <code>chaine</code> de droite à gauche à partir de sa dernière lettre ; puis un décalage est réalisé selon la comparaison effectuée. Cet algorithme permet de diminuer le nombre de comparaisons à effectuer.