

Exercice 1 (6 points)

Correction

Question	Niveau	Contenu	Solution
1	1	Passer de la représentation d'une base dans une autre.	01100001
2	1	Construire un tableau par compréhension.	[0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1]
3	2	Itérer sur les éléments d'un tableau. Construire une entrée de dictionnaire	<pre>14 if x in nb_occ: 15 nb_occ[x] = nb_occ[x] + 1 16 else: 17 nb_occ[x] = 1</pre>

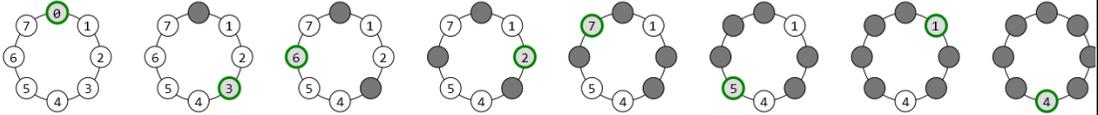
Question	Niveau	Contenu	Solution
4	2	Itérer sur les éléments d'un dictionnaire.	<pre> 14 for cle in dict.keys(): 15 if dict[cle] > valeur_max: 16 valeur_max = dict[cle] 17 cle_max = cle 18 return cle_max </pre>
5	1	Mise au point des programmes. Gestion des bugs.	L'erreur se situe à l'intersection de la première ligne et de la seconde colonne.
6	3	Utiliser des tableaux de tableaux pour représenter des matrices : notation $a[i][j]$.	<pre> 9 def erreur_colonne(mat): 10 for num_col in range(3): 11 if sum([mat[num_lig][num_col] for num_lig in 12 range(3)]) % 2 != 0: 13 return num_col </pre> <p>Autre version possible</p>

Question	Niveau	Contenu	Solution
7	1		<p>Le code du tableau le plus proche est 1110000, qui ne diffère que par son second bit du code reçu.</p> <p>Il correspond au mot 1000.</p>
8	2		<pre> 16 # Copie du code reçu créée par compréhension 17 code = [bit for bit in code_recu] 18 for indice in range(7): 19 # Inversion du bit d'indice courant 20 code[indice] = (code[indice] + 1) % 2 21 if code in liste_code_correct: 22 return code 23 else: 24 # Réinit. de bit d'indice courant 25 code[indice] = ((code[indice] + 1) % 2) </pre>
9	1	Arbres binaires : nœuds, racines, feuilles, sous- arbres gauches, sous-	$2^7 = 128$

Question	Niveau	Contenu	Solution
		arbres droits.	
10	3	Algorithmes sur les arbres binaires. Parcourir un arbre. Récursivité. Vocabulaire de la programmation objet : classes, attributs, méthodes, objets.	<pre> 11 if code[i] == 0: 12 return decode(arbre.gauche, code, i+1) 13 if code[i] == 1: 14 return decode(arbre.droit, code, i+1) </pre>

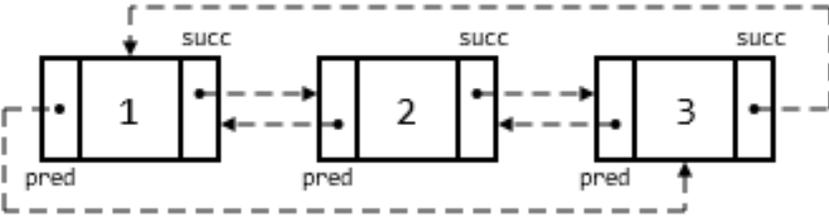
Exercice 2 (6 points)

Correction

Question	Niveau	Contenu	Solution
1	1	comprendre une situation/un problème	<p>On mange successivement les bonbons d'indices 0, 3, 6, 2, 7, 5, 1 et le dernier bonbon possède l'indice 4.</p>  <p>Figure 1. (correction)</p>
2	1	interpréter une erreur de programmation	<p>Une erreur de type <code>NameError</code> indique une variable de nom inconnu. Bob a saisi <code>true</code> que Python interprète comme un nom de variable. Il faut taper <code>True</code> afin d'utiliser le booléen.</p>
3	3	compléter une fonction Python simple	<pre> 1 def dernier(n): 2 collier = [True for i in range(n)] # création de la liste collier 3 indice = 0 4 collier[indice] = False 5 for etape in range(n-1): 6 nb_bonbons_vus = 0 7 while nb_bonbons_vus != 3: 8 indice += 1 9 if indice >= n: 10 indice = 0 </pre>

Question	Niveau	Contenu	Solution
			<pre> 11 if collier[indice]: 12 nb_bonbons_vus += 1 13 collier[indice] = False 14 return indice </pre>
4	1	connaître les structures de données	Une file est une structure de données linéaire suivant le modèle « <i>Premier entré, Premier sorti</i> ».
5	1	faire fonctionner un algorithme à la main	<p>Initialement la file est vide. On enfile ensuite les valeurs de [0, 1, 2, 3, 4]. 0 est en tête de la file et 4 à la queue.</p> <p>On défile donc successivement 0, puis 1 et enfin 2.</p> <p>Seules les deux dernières valeurs sont enfilées à nouveau.</p> <p>On a donc finalement : (Tête) 3 4 1 2 (Queue).</p>
6	2	compléter une fonction Python simple	<pre> def dernier_file(n): collier = File() for i in range(n): collier.enqueue(i) for tour in range(n - 1): collier.dequeue() collier.enqueue(collier.dequeue()) collier.enqueue(collier.dequeue()) return collier.dequeue() </pre>

Question	Niveau	Contenu	Solution
7	1	connaître le vocabulaire de la POO	Ces variables sont des <i>attributs</i> .
8	2	faire fonctionner un algorithme à la main	Le succ de zero est un. La variable a contient sa valeur, donc a = 1. Le pred du succ du succ de un est l'objet deux. La variable b contient sa valeur, donc b = 2.
9	2	compléter une fonction Python	<pre>def creer_collier(n): premier = Bonbon(0) actuel = premier for i in range(1, n): nouveau = Bonbon(i) actuel.succ = nouveau nouveau.pred = actuel actuel = nouveau actuel.succ = premier premier.pred = actuel return premier</pre>
10	2	faire fonctionner un	On travaille sur un collier de 4 bonbons, numérotés de 0 à 3.

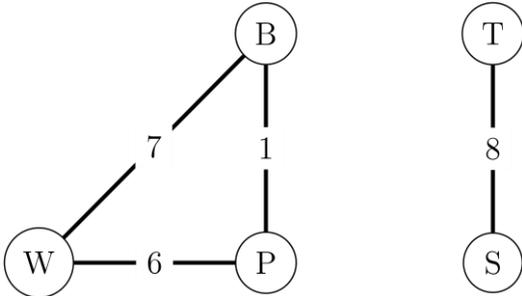
Question	Niveau	Contenu	Solution
		algorithme à la main	<p>Les opérations du code ont pour effet de supprimer le premier bonbon (de valeur 0). La variable <code>bonbon</code> est le <code>succ</code> de <code>premier</code> : il s'agit du bonbon de valeur 1.</p> <p>On a donc :</p>  <p>Figure 2. (correction)</p>
11	1	comprendre une situation/problème	Il s'agit de la proposition C : <code>bonbon.valeur == bonbon.succ.valeur</code> .
12	2	compléter une fonction Python complexe	<pre>def dernier_chaine(n): bonbon = creer_collier(n) while bonbon.valeur != bonbon.succ.valeur: bonbon.pred.succ = bonbon.succ bonbon.succ.pred = bonbon.pred bonbon = bonbon.succ.succ.succ return bonbon.valeur</pre>

Exercice 3 (8 points)

Correction

Question	Niveau	Contenu	Solution
1	1	Modèle relationnel	Il ne permet pas d'identifier de façon unique chaque enregistrement de la relation (sinon, cela ne permettrait qu'une seule réservation par vol, donc un seul passager par avion).
2	1	Modèle relationnel	<ul style="list-style-type: none">• le couple (id_vol, id_passager);• ou une proposition de nouvelle colonne id_reservation.
3	1	Modèle relationnel	Elle indique qu'une colonne dans une table fait référence à la clé primaire d'une autre table. Cela permet de créer des liens entre les données de différentes tables.
4	1	Langage SQL	AI0015, AI0258, AI0292
5	2	Langage SQL	<pre>SELECT ville FROM aeroport JOIN vol ON vol.aeroport_arr = id_aeroport WHERE aeroport_dep = 'CDG';</pre>
6	1	Langage SQL	<pre>UPDATE passager SET d_totale = 16 WHERE id_passager = 5;</pre>
7	2	Base de données	L'identifiant du vol n'est pas correct.

Question	Niveau	Contenu	Solution
		relationnelle et langage SQL	Remarque : il existe déjà ce qui va avoir pour effet le rejet de la requête puisqu'il s'agit d'une clé primaire. <pre>INSERT INTO vol (id_vol, aeroport_dep, aeroport_arr, distance) VALUES ('AI1000', 'CDG', 'YUL', 6);</pre>
8	1	Dictionnaire	10
9	2	Programmation et dictionnaire	<pre>1 def vol_direct(graphe, ville1, ville2): 2 return ville2 in graphe[ville1] 3 # ou : return ville1 in graphe[ville2]</pre>
10	3	Programmation et dictionnaire	<pre>1 def liste_villes_proches(graphe, ville, d_max): 2 resultat = list() 3 for voisine in graphe[ville]: 4 if graphe[ville][voisine] <= d_max: 5 resultat.append(voisine) 6 return resultat</pre>

Question	Niveau	Contenu	Solution
11	1	Graphes	 <pre> graph TD W((W)) --- 7 B((B)) W --- 6 P((P)) B --- 1 P T((T)) --- 8 S((S)) </pre> <p>Figure 2. Graphe non orienté pondéré correspondant au réseau aérien de la compagnie <i>Droidevant</i>.</p>
12	1	Graphes	Proposition A
13	1	Récurtivité	À la ligne 12 la fonction <code>parcours</code> s'appelle elle-même, la fonction <code>parcours</code> peut donc être qualifiée de fonction récursive.
14	3	Algorithmique - Récurtivité	La variable <code>visitees1</code> contient ['W', 'P', 'T', 'B', 'S']. La variable <code>visitees2</code> contient ['W', 'P', 'B'].
15	2	Parcours d'un graphe	Proposition C
16	2	Programmation	<pre> 1 def est_connexe(graphe): 2 """Vérifie si un graphe est connexe.""" 3 depart = ville_arbitraire(graphe) </pre>

Question	Niveau	Contenu	Solution
			<pre> 4 visitees = [] 5 parcours(graphe, visitees, depart) 6 return len(visitees) == len(graphe) </pre>
17	3	Algorithmique - Récursivité - Parcours d'un graphe	<pre> ['W', 'P', 'T', 'B'] 25 ['W', 'P', 'S', 'T', 'B'] 40 </pre>
18	3	Algorithmique - Récursivité - Parcours d'un graphe	<p>Cet appel affiche tous les chemins du graphe allant de ville à arrivee sans repasser par un même sommet ainsi que le poids de ces chemins, c'est-à-dire la somme des poids des arêtes de ces chemins.</p>