

Sujet : Représentation des données , types de bases

Exercice 1 (3 points) – Conversion de base 10 en base 2 et réciproquement

1.a – Donnez l'écriture en base 2 sur un octet du nombre entier 245.

11110101

1.b – Donnez l'écriture en base 2 sur un octet du nombre entier -93.

10100011

1.c – Donnez la valeur en base 10 du nombre binaire 01100110.

102

Exercice 2 (2 points) – Conversion de base 10 en base 16 et réciproquement

2.a – Donnez l'écriture en base 16 du nombre 245.

F5

2.b – Donnez la valeur en base 10 du nombre hexadécimal 24.

36

Sujet : Représentation des données , types construits

Exercice 3 (3 points) – Propriétés des tuples

Soit l'instruction suivante écrite en langage Python :

```
val = ('x', -6, 15.2, (7, 8.3, 'URL'), 'http')
```

3.a – Que s'affiche-t-il dans la console après avoir validé ceci : `print(val[2])` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

15.2

'val' est un tuple, or on accède aux données d'un tuple par l'index de leur position, sachant que l'index du premier élément est 0.

3.b - Que s'affiche-t-il dans la console après avoir validé ceci : `val[2] = 7.5` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche un message d'erreur. En effet 'val' est de type 'tuple', or les données d'un tuple ne sont pas modifiables.

3.c - Que s'affiche-t-il dans la console après avoir validé ceci : `ser = val[3] + (6, 'RAM')` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche (7, 8.3, 'URL, 6, 'RAM'). 'val' est de type tuple tout comme (6, 'RAM'), or les tuples sont concaténables.

Exercice 4 (3 points) – Propriétés des listes

Soit l'instruction suivante écrite en langage Python :

```
val = ['x', -6, 15.2, (7, 8.3, 'URL'), 'http']
```

4.a - Que s'affiche-t-il dans la console après avoir validé ceci : `print(len(val))` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche 5. La fonction len() permet de connaître le nombre d'éléments d'un itérable (tuple, liste, chaîne de caractères, dictionnaires)

4.b - Que s'affiche-t-il dans la console après avoir validé ceci : `print(val[5])` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche un message d'erreur ('index out of range'). En effet les 5 éléments de 'val' sont indexés de 0 à 4.

4.c - Que s'affiche-t-il dans la console après avoir validé ceci : `print(val.pop())` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche 'http'. 'pop()' est une méthode des liste qui par défaut en retire le dernier élément.

Exercice 5 (3 points) – Propriétés des dictionnaires

Soit l'instruction suivante écrite en langage Python :

```
dic = {'a': 7, 'b': 6, 'c': 2}
```

5.a - Que s'affiche-t-il dans la console après avoir validé ceci : `print(dic['a'] * 2)` ? Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche '14'. En effet la valeur associée à la clé 'a' est 7 que l'on multiplie par 2.

5.b – Que s'affiche-t-il dans la console après avoir validé ceci :

```
dic['b'] = 9
```

```
print(dic)
```

Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche {'a': 7, 'b': 9, 'c': 2}. La valeur 9 va remplacer celle qui est associée à la clé 'b'.

5.c – Que s'affiche-t-il dans la console après avoir validé ceci :

```
dic['e'] = 4
```

```
print(dic)
```

Expliquez en une seule phrase la réponse donnée en utilisant un vocabulaire approprié.

Il s'affiche {'a': 7, 'b': 9, 'c': 2, 'e': 4}. On ajoute la clé 'e' à la fin du dictionnaire et on lui associe la valeur 4.

Sujet : Langage et programmation : constructions de base

Exercice 6 (4 points) – Nommer des constructions

Attention ! Dans les questions qui suivent il ne s'agit pas d'expliquer à quoi servent les instructions mais d'indiquer le nom qui leur est donné d'une façon générale.

Soit le programme n°1 suivant écrit en langage Python :

```
1 def diviseurs(n):  
2     # n : nombre entier supérieur à 0  
3     cpt = 0  
4     for d in range(1, n + 1):  
5         if n % d == 0:  
6             cpt = cpt + 1  
7     return cpt
```

6.a - En informatique quel nom est donné à l'ensemble des instructions allant de la ligne 1 à la ligne 7 ?

Il s'agit d'une fonction.

6.b - En informatique quel nom est donné à une instruction comme celle de la ligne 3 ?

Il s'agit d'une affectation.

6.c - En informatique quel nom est donné à une instruction comme celle de la ligne 4 ?

Il s'agit d'une boucle bornée (ou boucle non conditionnelle, ou itération non conditionnelle).

6.d - En informatique quel nom est donné à une instruction comme celle de la ligne 5 ?

Il s'agit d'un test (ou expression conditionnelle).

6.e - En informatique quel nom est donné à une instruction comme celle de la ligne 6 ?

Il s'agit d'une incrémentation.

Soit le programme n°2 suivant écrit en langage Python :

```
1 def taille(n):  
2     # n : nombre entier supérieur à 0  
3     cpt = 0  
4     while n > 0:  
5         cpt = cpt + 1  
6         n = n // 2  
7     return cpt
```

6.f - En informatique quel nom est donné à une instruction comme celle de la ligne 4 ?

Il s'agit d'une boucle non bornée (ou boucle conditionnelle, ou itération conditionnelle).

6.g - En informatique quel nom est donné à ce qui écrit à la ligne 2 ?

Il s'agit d'un commentaire.

6.h - En informatique quel nom est donné au décalage de la ligne 2 par rapport à la ligne 1 ou au décalage de la ligne 5 par rapport à la ligne 4 par exemple ?

Il s'agit d'une indentation.

Sujet : Langage de haut niveau, assembleur et langage machine

Exercice 7 (4 points) - Le but de l'exercice est de comprendre ce que réalise un programme écrit dans un langage comparable à de l'assembleur.

Soit un CPU dont le jeu d'instructions est indiqué en annexe 1 (voir page située à la fin du sujet)

Soit le programme qui suit :

Adresse RAM	Instruction machine		Valeur contenue dans l'accumulateur
001	LOAD	\$ 2	2
002	SAVE	# 020	2
003	LOAD	\$ -5	-5
004	SAVE	# 021	-5
005	LOAD	\$ 0	0
006	SAVE	# 022	0
007	LOAD	\$ 16	16
008	SAVE	# 023	16
009	LOAD	# 022	0, 1, 2, 3
010	MUL	# 022	0, 1, 4, 9
011	MUL	# 020	0, 2, 8, 18
012	ADD	# 022	0, 3, 10, 21
013	ADD	# 021	-5, -2, 5, 16
014	CMP	# 023	1, 1, 1, 0
016	CAZ		
017	HLT		
018	INC	# 022	
019	JMP	# 009	
020		2	
021		-5	
022		0, 1, 2, 3	
023		16	

7.a – Écrire les nombres rangés aux adresses RAM 020 à 023 lors de l'exécution de ce programme . Si la valeur rangée à une adresse RAM change au cours de l'exécution de ce programme, écrire toutes ces valeurs dans leur ordre d'apparition et en les séparant par des virgules.

7.b – Écrire les valeurs contenues dans l'accumulateur lors de l'exécution de ce programme. Si plusieurs valeurs différentes sont placées successivement dans l'accumulateur par une même instruction machine, écrire toutes ces valeurs dans leur ordre d'apparition et en les séparant par des virgules.

Remarques :

>>> Ce programme s'arrête à un moment donné !

>>> Une adresse RAM et l'accumulateur changent au maximum 4 fois de valeurs.

>>> Ce programme permet de résoudre l'équation : $2x^2 + x - 5 = 16$

Sujet : Traitement de données en table

On dispose de données sur des films d'animation sortis en 2023.

(Source des données : <https://www.kaggle.com/datasets/gianinamariapetrascu/top-250-anime-2023>).

La capture d'écran qui suit présente une partie de ces données visualisées dans un tableur.

	A	B	C	D	E	F	G
	Titre	Genre	Studio	Type	Nombre_episodes	Duree_episode	Score
1	Ashita no Joe	Drama, Sports	Mushi Production	TV	79	25	8.29
2	Detective Conan Movie 06: The Phantom of Baker Street	Adventure, Mystery	TMS Entertainment	Movie	1	107	8.30
3	Tenki no Ko	Award Winning, Drama, Fantasy, Romance	CoMix Wave Films	Movie	1	112	8.30
4	Hibike! Euphonium 2	Drama	Kyoto Animation	TV	13	23	8.30
5	Blue Lock	Sports	8bit	TV	24	24	8.30
6	Kuroko no Basket 3rd Season	Sports	Production I.G	TV	25	24	8.30
7	Diamond no Ace: Second Season	Sports	Production I.G, Madhouse	TV	51	24	8.30
8	Evangeline: 2.0 You Can (Not) Advance	Action, Drama, Sci-Fi	Khara	Movie	1	112	8.31
9	Hunter x Hunter: Original Video Animation	Action, Adventure, Fantasy	Nippon Animation	OVA	8	24	8.31
10	Ano Hi Mita Hana no Namae wo Bokura ni Iba	Drama, Supernatural	A-1 Pictures	TV	11	22	8.31
11	Sakamichi no Apollon	Drama, Romance	Tezuka Productions, MAPPA	TV	12	22	8.31
12	Boku dake ga Inai Machi	Mystery, Supernatural, Suspense	A-1 Pictures	TV	12	23	8.31

Ces données ont été exportées dans un fichier 'csv' nommé 'animation.csv'.

Question 1 (0.5 point) - Que signifie 'csv' ?

Voici un aperçu du contenu de ce fichier ouvert dans un éditeur de texte.

```
Titre;Genre;Studio;Type;Nombre_episodes;Duree_episode;Score
Ashita no Joe;Drama, Sports;Mushi Production;TV;79;25;8.29
Detective Conan Movie 06: The Phantom of Baker Street;Adventure, Mystery;TMS Entertainment;Movie;1;107;8.30
Tenki no Ko;Award Winning, Drama, Fantasy, Romance;CoMix Wave Films;Movie;1;112;8.30
Hibike! Euphonium 2;Drama;Kyoto Animation;TV;13;23;8.30
```

Question 2 - (0.5 point) Quel est le caractère choisi comme séparateur de données ?

On veut importer ces données dans un programme écrit en langage Python sous la forme d'une liste de tuples.

Question 3 - (2 points) A partir du programme qui suit :

```
1 def charge_data(nom_fichier):
2     r = open(nom_fichier, mode="____", encoding="_____")
3
4     # Lecture de la première ligne du fichier csv et affectation à une variable
5     libelles = r.readline()
6     # suppression du caractère de retour à la ligne situé à la fin
7     libelles = libelles.replace("\n", "")
8     # découpage de la chaîne de caractères en éléments d'une liste
9     libelles = libelles.split(____)
10    #
11    libelle = tuple(libelles)
12
13    table = r._____
14    for index in range(len(table)):
15        table[index] = table[index].replace("\n", "")
16        table[index] = table[index].split(____)
17        table[index] = tuple(table[index])
18
19    r.close()
20    return libelles, table
--
```

Recopier et compléter lisiblement sur la copie :

- l'instruction de la ligne 2,
- le commentaire de la ligne 10 (indiquer le rôle de la ligne 11)

- l'instruction de la ligne 13,
- les instructions des lignes 9 et 16

On exécute l'instruction suivante :

```
libelles, table = charge_data("animation.csv")
```

Voici ce qu'on peut lire dans l'explorateur de variables :

Nom	Type	Taille	Valeur
libelles	list	7	['Titre', 'Genre', 'Studio', 'Type', 'Nombre_episodes', 'Duree_episode ...
table	list	246	[('Ashita no Joe', 'Drama, Sports', 'Mushi Production', 'TV', '79',

Une table est constituée de lignes et de colonnes.

Question 4a - (0.5 point) Quel autre mot peut-être employé pour parler des lignes ou enregistrements d'une table ?

Question 4b - (0.5 point) Quel autre mot peut-être employé pour parler des colonnes ou descripteurs d'une table ?

La variable 'libelles' contient les intitulés de chaque colonne qui indiquent la nature des données contenues dans chacune des colonnes.

Question 5a - (0.5 point) Combien de colonnes comporte la table des films d'animation ?

Chacun des tuples de la liste 'table' contient les données pour un film d'animation.

Question 5b - (0.5 point) Combien de films comporte la table des films d'animation ?

On voudrait afficher dans la console les données du 143^e film présent dans 'table'.

Question 6a - (1 point) Quelle instruction faut-il écrire ?

On voudrait afficher dans la console le nombre d'épisodes du 25^e film de la table.

Question 6b - (1 point) Quelle instruction faut-il écrire ?

Question 7 - (1 point) Écrire un programme en langage Python qui permet de connaître le nombre de films dont le 'Type' est 'Movie'.

Annexe 1 – Jeu d'instructions d'un CPU

Opération		Opérande	Description
Code	Mnémonique		
100	LOAD	# m	Charger dans l'accumulateur la valeur située en RAM à l'adresse m.
110	LOAD	\$ v	Charger la valeur v dans l'accumulateur.
150	SAVE	# m	Écrit la valeur de l'accumulateur à l'adresse m en RAM.
200	ADD	# m	Ajoute à la valeur qui se trouve dans l'accumulateur celle située en RAM à l'adresse m.
210	SUB	# m	Retire à la valeur qui se trouve dans l'accumulateur celle située en RAM à l'adresse m.
250	INC	# m	Incrémente la valeur située en RAM à l'adresse m.
260	DEC	# m	Décrémente la valeur située en RAM à l'adresse m.
220	MUL	# m	Multiplie la valeur qui se trouve dans l'accumulateur avec celle située en RAM à l'adresse m.
230	QUO	# m	Réalise la division entière de la valeur située en RAM à l'adresse m par celle qui se trouve dans l'accumulateur, et place le quotient dans l'accumulateur.
240	MOD	# m	Réalise la division entière de la valeur située en RAM à l'adresse m par celle qui se trouve dans l'accumulateur, et place le reste dans l'accumulateur.
300	CPZ	# m	Compare la valeur située en RAM à l'adresse m avec zéro. Exécution l'opération qui suit en cas d'égalité, sinon passe cette opération et continue à la suivante.
310	CAZ		Compare la valeur de l'accumulateur avec zéro. Exécute l'opération qui suit en cas d'égalité, sinon passe cette opération et continue à la suivante.
320	CMP	# m	Compare la valeur (val) située en RAM à l'adresse m avec celle de l'accumulateur (acc) : place 1 dans l'accumulateur si val > acc ; place -1 dans l'accumulateur si acc > val ; place 0 dans l'accumulateur si acc = val.
400	JMP	# m	Poursuit le programme à l'instruction située à l'adresse m en RAM.
500	HLT		Arrête le programme.