

BACCALAURÉAT GÉNÉRAL

ÉPREUVE D'ENSEIGNEMENT DE SPÉCIALITÉ

SESSION 2024

NUMÉRIQUE ET SCIENCES INFORMATIQUES

Janvier 2024

Durée de l'épreuve : 3 heures 30

L'usage de la calculatrice et du dictionnaire n'est pas autorisé.

Dès que ce sujet vous est remis, assurez-vous qu'il est complet.
Ce sujet comporte xx pages numérotées de 1/6 à 6/6.

Le candidat traite 3 exercices

Exercice 1

/12

/6

Thème abordé : programmation objet en langage Python

Un fabricant de brioches décide d'informatiser sa gestion des stocks. Il écrit pour cela un programme en langage Python. Une partie de son travail consiste à développer une classe Stock dont la première version est la suivante :

```
class Stock:
    def __init__(self):
        self.qt_farine = 0 # quantité de farine initialisée à 0 g
        self.nb_oeufs = 0 # nombre d'œufs (0 à l'initialisation)
        self.qt_beurre = 0 # quantité de beurre initialisée à 0 g
```

1. Écrire une méthode `ajouter_beurre(self, qt)` qui ajoute la quantité `qt` de beurre à un objet de la classe Stock.

/2

On admet que l'on a écrit deux autres méthodes `ajouter_farine` et `ajouter_oeufs` qui ont des fonctionnements analogues.

2. Écrire une méthode `afficher(self)` qui affiche la quantité de farine, d'œufs et de beurre d'un objet de type Stock. L'exemple ci-dessous illustre l'exécution de cette méthode dans la console :

```
>>> mon_stock = Stock()
>>> mon_stock.afficher()
farine: 0
oeuf: 0
beurre: 0
>>> mon_stock.ajouter_beurre(560)
>>> mon_stock.afficher()
farine: 0
oeuf: 0
beurre: 560
```

/2

3. Pour faire une brioche, il faut 350 g de farine, 175 g de beurre et 4 œufs. Écrire une méthode `stock_suffisant_brioche(self)` qui renvoie un booléen : VRAI s'il y a assez d'ingrédients dans le stock pour faire une brioche et FAUX sinon.

/2

4. On considère la méthode supplémentaire `produire(self)` de la classe `Stock` donnée par le code suivant :

```
def produire(self):
    res = 0
    while self.stock_suffisant_brioche():
        self.qt_beurre = self.qt_beurre - 175
        self.qt_farine = self.qt_farine - 350
        self.nb_oeufs = self.nb_oeufs - 4
        res = res + 1
    return res
```

On considère un stock défini par les instructions suivantes :

```
>>> mon_stock=Stock()
>>> mon_stock.ajouter_beurre(1000)
>>> mon_stock.ajouter_farine(1000)
>>> mon_stock.ajouter_oeufs(10)
```

a. On exécute ensuite l'instruction

```
>>> mon_stock.produire()
```

Quelle valeur s'affiche dans la console ? Que représente cette valeur ?

/2

b. On exécute ensuite l'instruction

```
>>> mon_stock.afficher()
```

Que s'affiche-t-il dans la console ?

/1,5

5. L'industriel possède n lieux de production distincts et donc n stocks distincts.

On suppose que ces stocks sont dans une liste dont chaque élément est un objet de type `Stock`. Écrire une fonction Python `nb_brioche(liste_stocks)` possédant pour unique paramètre la liste des stocks et renvoie le nombre total de brioches produites.

/2,5

Exercice 2

/16

/8

Cet exercice porte sur les bases de données

Dans le cadre d'une étude sur le réchauffement climatique, un centre météorologique rassemble des données. On considère que la base de données contient deux relations (tables). La relation Centres qui contient l'identifiant des centres météorologiques, la ville, la latitude, la longitude et l'altitude du centre. La relation Mesures qui contient l'identifiant de la mesure, l'identifiant du centre, la date de la mesure, la température, la pression et la pluviométrie mesurées.

Le schéma relationnel de la relation Centres est le suivant :

Centres(id_centre: INT, nom_ville: VARCHAR, latitude: FLOAT, longitude: FLOAT, altitude: FLOAT)

Le schéma relationnel de la relation Mesures est le suivant :

Mesures(id_mesure: INT, id_centre: INT, date: DATE, temperature: FLOAT, pression: INT, pluviometrie: FLOAT)

Relation Centres

id centre	nom ville	latitude	longitude	altitude
213	Amiens	49.894	2.293	60
138	Grenoble	45.185	5.723	550
263	Brest	48.388	-4.49	52
185	Tignes	45.469	6.909	2594
459	Nice	43.706	7.262	260
126	Le Puy-en-Velay	45.042	3.888	744
317	Gérardmer	48.073	6.879	855

Relation Mesures

id mesure	id centre	date	temperature	pression	pluviometrie
1566	138	2021-10-29	8.0	1015	3
1568	213	2021-10-29	15.1	1011	0
2174	126	2021-10-30	18.2	1023	0
2200	185	2021-10-30	5.6	989	20
2232	459	2021-10-31	25.0	1035	0
2514	213	2021-10-31	17.4	1020	0
2563	126	2021-11-01	10.1	1005	15
2592	459	2021-11-01	23.3	1028	2
3425	317	2021-11-02	9.0	1012	13
3430	138	2021-11-02	7.5	996	16
3611	263	2021-11-03	13.9	1005	8
3625	126	2021-11-03	10.8	1008	8

1.
a. Proposer une clé primaire pour la relation Mesures. Justifier votre choix.

/1

- b. Avec quel attribut peut-on faire une jointure entre la relation Centres et la relation Mesures ?

/1

c. Citer une contrainte d'intégrité de domaine de la relation Centres .

/0,5

2.

a. Qu'affiche la requête suivante ?

```
SELECT *  
FROM Centres  
WHERE altitude>500;
```

/1

b. On souhaite récupérer le nom de la ville des centres météorologiques situés à une altitude comprise entre 700m et 1200m. Ecrire la requête SQL correspondante.

/3

c. On souhaite récupérer la liste des longitudes et des noms des villes des centres météorologiques dont la longitude est supérieure à 5. La liste devra être triée par ordre alphabétique des noms de ville. Ecrire la requête SQL correspondante.

/3

3.

a. Donner la requête permet d'afficher l'ensemble des données (id_mesure, id_centre, date, température, pression et pluviométrie) des mesures effectuées le 30 octobre 2021.

/1,5

b. Écrire une requête SQL permettant d'ajouter une mesure prise le 8 novembre 2021 dans le centre numéro 138, où la température était de 11°C, la pression de 1013 hPa et la pluviométrie de 0mm. La donnée dont l'attribut est id_mesure aura pour valeur 3650.

/2

4.

a. Expliquer ce que renvoie la requête SQL suivante ?

```
SELECT *  
FROM Centres  
WHERE latitude = (SELECT MIN(latitude) FROM Centres);
```

/1

b. Écrire une requête SQL donnant la liste des villes dans lesquelles on a enregistré une température inférieure à 10°C en octobre 2021. On utilisera le mot clé DISTINCT afin d'éviter d'avoir des doublons. On rappelle que l'on peut utiliser les opérateurs de comparaison avec les dates.

/2

Exercice 3

/12 /6

Cet exercice porte sur les structures de données (dictionnaires).

La cryptographie est un ensemble de techniques permettant de chiffrer un message.

Une technique de cryptographie consiste à mélanger les lettres d'un alphabet et à réécrire le message avec ces permutations. En Python, on peut créer un dictionnaire dans lequel les clés sont les lettres de l'alphabet et les valeurs sont celles de l'alphabet mélangé.

Par exemple, si l'alphabet contient les 4 lettres A, B, C et D, et si le dictionnaire de l'alphabet mélangé est $\alpha = \{ "A": "B", "B": "D", "C": "A", "D": "C" \}$, la chaîne de caractères "BAC" sera chiffrée "DBA".

Un tel dictionnaire sera appelé **dictionnaire de chiffrement**.

1. On souhaite chiffrer un message écrit avec l'alphabet A, B, C, D, E, F, G à l'aide du dictionnaire `alpha = {"A": "B", "B": "D", "C": "A", "D": "C", "E": "F", "F": "G", "G": "E"}`

a. Quelle est la valeur associée à la clé "D" ? En Python, comment l'obtenir ? /1

b. Chiffrer la chaîne de caractères "BAGAGE" avec le dictionnaire `alpha`. /0,5

2. On considère qu'un mot est une chaîne de caractères (un objet de type `str`) écrite uniquement avec les 26 lettres de l'alphabet en majuscule. Par exemple, "ARBRE" est un mot et "L'ARBRE !" n'est pas un mot à cause des caractères : "'", " " (espace) et "!". Écrire une fonction `chiffrer(mot, alpha)` qui prend en paramètres `mot` un mot et `alpha` un dictionnaire de chiffrement, et qui renvoie une chaîne de caractères chiffrée avec le dictionnaire de chiffrement `alpha`. /4

3. On souhaite déchiffrer un mot chiffré avec cette méthode.

a. Si un mot est chiffré avec le dictionnaire de chiffrement `alpha = {"A": "B", "B": "D", "C": "A", "D": "C", "E": "F", "F": "G", "G": "E"}`, donner un dictionnaire permettant de le déchiffrer. /1

b. Ecrire une fonction en Python appelée `dico_dechiffrement(dico)` qui prend en paramètre `dico` un dictionnaire de chiffrement et qui renvoie un dictionnaire permettant le déchiffrement. On pourra compléter le code ci-dessous ou proposer une autre solution :

```
def dico_dechiffrement(dico):  
    nouveau = {}  
    for lettre in dico :  
        code = dico[.....]  
        nouveau[.....] = .....  
    return nouveau
```

 /1,5

c. Ecrire une fonction `dechiffre(mot, dico)` qui reçoit un mot chiffré et un dictionnaire de chiffrement et renvoie le mot décodé. On utilisera les fonctions écrites dans les questions précédentes. /4