

RÉPUBLIQUE TUNISIENNE ☆☆☆ Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT		SESSION 2025	
	ÉPREUVE PRATIQUE D'INFORMATIQUE			
	Section : Économie et Gestion			
	Coefficient de l'épreuve: 0.5		Durée: 1h	

Le sujet comporte 4 pages numérotées de 1/4 à 4/4

Important : Dans le répertoire **Bac2025** situé sur la racine du disque **C** de votre poste, créez un dossier de travail portant votre numéro d'inscription (6 chiffres) et dans lequel vous devez enregistrer, au fur et à mesure, tous les fichiers solutions de ce sujet.

Afin de gérer la location des tentes dans un centre de camping et de garantir la sécurité et la facilité de gestion des accès, le centre propose des bracelets connectés. Un bracelet est automatiquement désactivé dès que la durée d'un camping est écoulée. Pour ce faire, le responsable exploite la base de données simplifiée "**GestionCamping**" décrite par la représentation textuelle suivante :

CAMPEUR (IdCamp, NomPrenCamp, TelCamp)

TYPETENTE (RefType, Libelle, Capacite, Prix)

TENTE (CodeT, NomT, RefType#)

BRACELET (IdCamp#, CodeT#, DateArr, NbJours)

Sachant que :

- **CAMPEUR** est une table contenant les informations relatives aux différents campeurs.
- **TENTE** est une table contenant les informations relatives aux tentes du centre de camping.
- **TYPETENTE** est une table contenant les informations relatives aux différents types de tentes du centre de camping.
- **BRACELET** est une table contenant les informations relatives aux différents séjours des campeurs.

Soit la description des colonnes des tables de cette base de données :

Nom	Description	Type	Taille	Format	Contrainte
IdCamp	Identifiant d'un campeur.	Texte court	3		
NomPrenCamp	Nom et prénom d'un campeur.	Texte court	30		
TelCamp	Numéro du téléphone d'un campeur.	Texte court	8		Null interdit
RefType	Référence attribuée à un type de tente.	Texte court	2		
Libelle	Libellé décrivant un type de tente.	Texte court	20		Null interdit
Capacite	Nombre maximal de personnes par tente.	Numérique	Entier		Entre 1 et 10
Prix	Prix d'une nuitée par type de tente.	Numérique	Réel simple		3 décimales
CodeT	Code d'une tente.	Texte court	3		
NomT	Nom d'une tente.	Texte court	20		
DateArr	Date d'arrivée au centre de camping.	Date/Heure		Date, abrégé	
NbJours	Le nombre de jours passés dans le camping.	Numérique	Entier		>=3 et <=90 sinon un message d'erreur doit être affiché

A) À l'aide du logiciel de gestion de base de données disponible :

- 1) Créer, dans votre dossier de travail, la base de données à nommer "**GestionCamping**".
- 2) Créer les tables et les relations relatives à cette base de données tout en respectant les types, les tailles, les formats et les contraintes cités dans la description ci-dessus.

3) Remplir les différentes tables par les données représentées dans les tableaux suivants :

CAMPEUR		
IdCamp	NomPrenCamp	TelCamp
C01	Ramzi SBAIL	82123123
C02	Rami GHANMI	83111222
C03	Raouf BADI	82665544
C04	Rim SALHI	82333333

TYPETENTE			
RefType	Libelle	Capacite	Prix
A	Dôme XXL	6	125,5
B	Tipi vaste	9	405
C	Tunnel	8	337,5
D	Safari	4	120,5

TENTE		
CodeT	NomT	RefType
T01	Les vagues	D
T02	Les plongeurs	B
T03	Les sirènes	A
T04	AquaCamp	C
T05	Les crabes	A
T06	Etoile de mer	A
T07	Les dauphins	A
T08	Terre & mer	D

BRACELET			
IdCamp	CodeT	DateArr	NbJours
C01	T03	03/07/2024	6
C01	T06	14/05/2025	8
C02	T07	12/07/2024	4
C03	T01	10/08/2024	3
C04	T02	17/02/2025	3
C04	T03	20/07/2024	10
C04	T03	25/08/2025	5

4) Créer les requêtes suivantes :

R1 : Afficher les données (**NomPrenCamp**, **NomT**, **Capacite**, **Prix**, **DateArr**, **NbJours**, **RefType**) relatives aux bracelets des campeurs par ordre croissant des dates d'arrivée.

R2 : Diminuer de 15%, les prix des tentes de type "B" et de type "C".

R3 : Afficher le nombre total des jours d'occupation pour une tente de code donné.

5) Exporter, dans votre dossier de travail et sous le nom "**Camping.csv**", le résultat de la requête "**R1**" au format **csv** (Utiliser un **point-virgule** comme séparateur de champs, un **point** comme séparateur décimal, la page de codes **Unicode** (UTF-8) et inclure les noms des champs sur la première ligne).

B) À l'aide de l'environnement de développement disponible et en exploitant la bibliothèque Pandas :

- 1) Créer un fichier et l'enregistrer sous le nom "**Camping**" dans votre dossier de travail.
- 2) Dans ce fichier, choisir les méthodes et les propriétés adéquates à partir de l'annexe (pages 3 et 4), pour écrire les scripts permettant :

a) d'importer le contenu du fichier "**Camping.csv**" et de le stocker dans un DataFrame à nommer "**camp**",

b) d'afficher les dimensions du DataFrame "**camp**",

c) de modifier, à la ligne 5 du DataFrame "**camp**", le nom de la tente "**Etoile de mer**" par le nom "**Corail**", puis d'afficher cette ligne,

d) d'afficher le nombre total de jours passés dans le centre de camping par "**Rim SALHI**",

e) de calculer, dans une nouvelle colonne à nommer "**PrixTotal**", le prix total à payer par chaque campeur puis d'afficher le contenu du DataFrame "**camp**", sachant que :

$$\text{PrixTotal} = \text{Prix} \times \text{NbJours}$$

f) de trier le DataFrame "**camp**" dans un nouveau DataFrame à nommer "**camp_tri**" selon l'ordre croissant du nombre de jours puis d'afficher le DataFrame trié.

Grille d'évaluation	Partie	A (13 points)					B (7 points)						
	Question	1	2	3	4	5	1	2.a.	2.b.	2.c.	2.d.	2.e.	2.f.
	Note	0.5	3.75	3	4.5	1.25	0.5	1.25	0.25	1	1.25	1.25	1.5

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
<i>Création d'un DataFrame</i>	▪ <code>IdDataFrame = pandas.DataFrame ({"Id_Colonne1":["Val1",..., "ValN"],..., "Id_ColonneM":["Val1,...,ValN"]})</code>
<i>Importation des données dans un DataFrame</i>	▪ <code>IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille")</code> ▪ <code>IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")</code>
<i>Manipulation d'un DataFrame</i>	▪ <code>IdDataFrame.shape</code> ou bien <code>print (IdDataFrame.shape)</code> ▪ <code>IdDataFrame.size</code> ou bien <code>print (IdDataFrame.size)</code> ▪ <code>IdDataFrame.info ()</code> ou bien <code>print (IdDataFrame.info ())</code> ▪ <code>IdDataFrame.columns</code> ou bien <code>print (IdDataFrame.columns)</code> ▪ <code>Resultat = IdDataFrame.rename (columns = {"NomColonne1": "NouveauNomColonne1", "NomColonne2": "NouveauNomColonne2", ... })</code> ▪ <code>Resultat = IdDataFrame.drop ([N° Ligne1, ... , N° LigneN])</code> ▪ <code>Resultat = IdDataFrame.drop (IdDataFrame.index [N° LigneInitial : N° LigneFinal])</code> ▪ <code>Resultat = IdDataFrame.drop (columns = ["NomColonne1",..., "NomColonneN"])</code>
<i>Affichage des données d'un DataFrame</i>	▪ <code>IdDataFrame</code> ou bien <code>print (IdDataFrame)</code> ▪ <code>IdDataFrame ["NomColonne"]</code> ou bien <code>print (IdDataFrame ["NomColonne"])</code> ▪ <code>IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]]</code> ou bien <code>print (IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]])</code> ▪ <code>IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])</code> ▪ <code>IdDataFrame.loc [N° Ligne]</code> ou bien <code>print (IdDataFrame.loc [N° Ligne])</code> ▪ <code>IdDataFrame.head (n)</code> ou bien <code>print (IdDataFrame.head (n))</code> ▪ <code>IdDataFrame.tail (n)</code> ou bien <code>print (IdDataFrame.tail (n))</code> ▪ <code>IdDataFrame.loc [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame.loc [N° LigneInitial : N° LigneFinal])</code> ▪ <code>IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , ...]]</code> ou bien <code>print (IdDataFrame.loc [[N° Ligne1 , N° Ligne2 , ...]])</code> ▪ <code>IdDataFrame.loc [N° Ligne , "NomColonne"]</code> ou bien <code>print (IdDataFrame.loc [N° Ligne , "NomColonne"])</code>

Catégorie	Syntaxe
<i>Modification et ajout des données dans un DataFrame</i>	▪ <code>IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule)</code> ▪ <code>IdDataFrame.loc [N° ligne] = [Liste_Valeur]</code> ▪ <code>IdDataFrame ["Id_Colonne"] = Valeur (ou Formule)</code>
<i>Nettoyage d'un DataFrame</i>	▪ <code>IdDataFrame.drop_duplicates ()</code> ▪ <code>IdDataFrame.dropna ()</code> ▪ <code>IdDataFrame.dropna (axis = 1)</code>
<i>Les fonctions statistiques</i>	Liste des fonctions : <code>mean ()</code> – <code>min ()</code> – <code>max ()</code> – <code>sum ()</code> – <code>count ()</code> Syntaxe : <code>IdDataFrame ["NomColonne"].NomFonction()</code> ou bien <code>print (IdDataFrame ["NomColonne"].NomFonction())</code> Utilisation des fonctions avec groupement : Regroupement des données selon une colonne : <code>IdDataFrame.groupby ("Id_Colonne_à_regrouper") ["Id_Colonne_à_aggréger"].NomFonction()</code> ou bien <code>print (IdDataFrame.groupby ("Id_Colonne_à_regrouper") ["Id_Colonne_à_aggréger"].NomFonction())</code> Regroupement des données selon plusieurs colonnes : <code>IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]].groupby ("Id_Colonne1_Critère").NomFonction()</code> ou bien <code>print (IdDataFrame [["Id_Colonne1", "Id_Colonne2", ...]].groupby ("Id_Colonne1_Critère").NomFonction())</code>
<i>Affichage des données d'un DataFrame selon une ou plusieurs conditions</i>	▪ <code>IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur]</code> ▪ <code>IdDataFrame [(IdDataFrame["NomColonne"]OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]</code>
<i>Tri des éléments d'un DataFrame</i>	<code>DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1", "NomColonne2" , ...] ascending = [True/False , True/False , ...])</code>
<i>Création d'un graphique</i>	▪ <code>IdDataFrame.plot.bar (x = "NomColonneAbscisse", y = "NomColonneOrdonnée", title = "TitreGraphique", color = "CouleurGraphique")</code> ▪ <code>IdDataFrame.plot.line (x = "NomColonneAbscisse", y = "NomColonneOrdonnée", title = "TitreGraphique", color = "CouleurGraphique")</code> ▪ <code>alias_matplotlib.show ()</code>