

RÉPUBLIQUE TUNISIENNE ----- MINISTÈRE DE L'ÉDUCATION	EXAMEN DU BACCALAURÉAT SESSION 2026	
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Section : Économie et Gestion	
	Coefficient de l'épreuve : 0.5	Durée : 1 heure

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

Important : Dans le répertoire **Bac2026** 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 à ce sujet.

Afin d'assurer la livraison des colis, une société de livraison à domicile exploite des drones pour servir ses clients. Pour gérer ses livraisons, le responsable du service informatique exploite une base de données simplifiée intitulée "**BD_LivraisonDrone**" décrite par la représentation textuelle suivante :

DRONE (IdDrone, ChargeMax)

DESTINATION (IdDest, Latitude, Longitude, Tel)

MISSION (IdDest#, IdDrone#, DateMission, HeureMission, Distance, PoidsColis)

Sachant que pour une mission, un drone assure la livraison d'un seul colis vers une seule destination.

Soit la description des colonnes des tables de la base de données "**BD_LivraisonDrone**" :

Nom	Description	Type	Taille	Format	Contrainte
IdDrone	Identifiant d'un drone.	Texte court	3		
ChargeMax	Charge maximale que peut transporter un drone, exprimée en Kg.	Numérique	Entier		Supérieure ou égale à 10 et inférieure à 30
IdDest	Identifiant d'une destination.	Texte court	4		
Latitude	Coordonnée géographique d'une destination par rapport à l'équateur.	Texte court	7		
Longitude	Coordonnée géographique d'une destination par rapport au méridien de Greenwich.	Texte court	7		
Tel	Numéro de téléphone d'un client réceptionnaire d'un colis.	Texte court	8		
DateMission	Date d'une mission de livraison.	Date/Heure		Date, abrégé	
HeureMission	Heure d'une mission de livraison.	Date/Heure		Heure, abrégé	
Distance	Distance survolée par un drone vers une destination exprimée en Km.	Numérique	Entier		Supérieure à 1
PoidsColis	Poids d'un colis exprimé en Kg (supposé inférieur ou égal à la charge maximale d'un drone).	Numérique	Entier		Supérieur à 0

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 "**BD_LivraisonDrone**".
- 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 :

Table : DRONE	
IdDrone	ChargeMax
D01	20
D02	15
D03	25
D04	20

Table : DESTINATION			
IdDest	Latitude	Longitude	Tel
DS01	36.819N	10.166E	66222333
DS02	36.878N	10.324E	66111555
DS03	36.859N	10.330E	66999444
DS04	36.810N	10.095E	66666666

Table : MISSION					
IdDest	IdDrone	DateMission	HeureMission	Distance	PoidsColis
DS01	D01	12/01/2026	14:00	3	7
DS01	D02	14/02/2026	9:00	3	5
DS02	D03	14/02/2026	10:00	11	25
DS03	D04	01/03/2026	17:00	14	18
DS04	D02	18/04/2026	17:00	8	9
DS02	D02	23/04/2026	10:00	11	13
DS01	D04	26/04/2026	13:00	3	20

4) Créer les requêtes suivantes :

R1 : Afficher les missions de livraison des colis (**IdDest**, **IdDrone**, **ChargeMax**, **DateMission**, **HeureMission**, **Distance**, **PoidsColis**) triées par ordre chronologique selon la date puis l'heure de la mission.

R2 : Afficher, pour un drone d'identifiant donné, la liste des missions de livraison effectuées.

R3 : Afficher, pour chaque drone, la distance totale parcourue.

R4 : Afficher, la liste des missions de livraison des colis (**IdDest**, **Latitude**, **Longitude**, **DateMission**, **HeureMission**, **Distance**) effectuées pendant le mois d'avril 2026.

5) Exporter, dans votre dossier de travail et sous le nom "**Livraison.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 "**Missions**" dans votre dossier de travail.

2) En se référant à l'annexe, utiliser les méthodes et les propriétés adéquates, pour écrire dans le fichier **Missions** les scripts permettant :

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

b) d'afficher le DataFrame **drones**,

c) d'ajouter, à la fin du DataFrame **drones**, la mission :

IdDest	IdDrone	ChargeMax	DateMission	HeureMission	Distance	PoidsColis
DS04	D01	20	14/05/2026	11:00	8	17

d) d'afficher les missions de livraison dont le poids du colis est identique à la charge maximale du drone,

e) de calculer, dans une nouvelle colonne à nommer "**Frais**", le coût de la livraison d'un colis, sachant que : **Frais** = **PoidsColis** * 0.7 + **Distance** * 0.5

f) de trier, dans un nouveau DataFrame à nommer "**dronestri**", les données du DataFrame **drones** par ordre croissant des identifiants des destinations puis par ordre décroissant des frais de livraison et 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	2.25	6	1.25	0.5	1.25	0.25	1	1.25	1	1.75

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> <code>IdDataFrame.groupby (["Id_Colonne "]).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>