

RÉPUBLIQUE TUNISIENNE ☆☆☆ Ministère de l'Éducation	EXAMEN DU BACCALAURÉAT	SESSION 2023
	ÉPREUVE PRATIQUE D'INFORMATIQUE	
	Section : Économie et Gestion	
	Coefficient de l'épreuve : 0.5	Durée : 1h

Le sujet comporte 3 pages numérotées de 1 sur 3 à 3 sur 3

Important : Dans le répertoire **Bac2023** 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 les victoires des courses de la **Formula1**, on se propose d'utiliser la base de données simplifiée intitulée "**GestionFormula1**" décrite par le schéma textuel suivant :

MARQUEAUTO (CodeMarq, LibMarq)

PILOTE (IdPilote, NomPilote, DateNais)

CIRCUIT (CodeCircuit, NomCir, PaysCir, NbTours, LongCir)

VICTOIRE (IdPilote#, CodeCircuit#, Annee, TempsRealise, CodeMarq#)

Sachant que :

- **MARQUEAUTO** est une table contenant les informations relatives aux différentes marques d'automobile (Mercedes, Ferrari, etc.).
- **PILOTE** est une table contenant les informations relatives aux différents pilotes.
- **CIRCUIT** est une table contenant les informations relatives aux différents circuits du Formula1.
- **VICTOIRE** est une table contenant les informations relatives aux vainqueurs dans les différentes courses faites.

Soit la description des colonnes des tables de la base de données "**GestionFormula1**" :

Nom	Description	Type	Taille	Format	Contrainte
CodeMarq	Code d'une marque d'automobile	Texte	4		
LibMarq	Marque d'une automobile	Texte	100		Null interdit
IdPilote	Identifiant d'un pilote	Texte	4		
NomPilote	Nom et prénom d'un pilote	Texte	100		Null interdit
DateNais	Date de naissance d'un pilote	Date/Heure		Date, abrégé	
CodeCircuit	Code d'un circuit	Texte	4		
NomCir	Nom d'un circuit	Texte	100		Null interdit
PaysCir	Pays d'un circuit	Texte	100		Null interdit
NbTours	Nombre de tours d'un circuit	Numérique	Entier		Null interdit et >0
LongCir	Longueur d'un circuit exprimée en mètre	Numérique	Entier		Null interdit et >0
Annee	Année d'une compétition	Numérique	Entier		Null interdit
TempsRealise	Temps réalisé dans une course par le pilote vainqueur	Date/Heure		Heure, complet	Null interdit

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 "**GestionFormula1**".
- 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:

MARQUEAUTO	
CodeMarq	LibMarq
M001	Ferrari
M002	Mercedes
M003	Alpine
M004	Red Bull Racing

PILOTE		
IdPilote	NomPilote	DateNais
P001	Lewis HAMILTON	07/01/1985
P002	Sergio PEREZ	26/01/1990
P003	Esteban OCAN	17/09/1996
P004	Max VERSTAPPEN	30/09/1997

CIRCUIT				
CodeCircuit	NomCir	PaysCir	NbTours	LongCir
C001	Circuit de Sakhir	Bahrein	57	5412
C002	Circuit international de Losail	Qatar	57	5380
C003	Hungaroring	Hongrie	70	4381
C004	Circuit Yas Marina	Emirats	58	5281

VICTOIRE				
IdPilote	CodeCircuit	Annee	TempsRealise	CodeMarq
P001	C001	2021	1:32:03	M002
P001	C002	2021	1:24:28	M002
P002	C004	2021	2:13:36	M004
P003	C003	2021	2:04:43	M003
P001	C001	2022	1:27:23	M002

4) Créer les requêtes suivantes :

R1 : Afficher la liste des victoires (**NomPilote**, **LibMarq**, **NomCir**, **PaysCir**, **Annee**, **TempsRealise**).

R2 : Afficher la liste des circuits (**NomCir**, **PaysCir**, **NbTours**, **LongCir**) dans lesquels "Lewis HAMILTON" a réalisé des victoires pendant l'année **2021**.

R3 : Afficher, pour chaque pilote vainqueur en **2021**, son nom et son prénom (**NomPilote**) et sa date de naissance, ainsi que le nombre total de ses victoires.

5) Exporter, dans votre dossier de travail sous le nom "**circuit.csv**", le contenu de la table "**CIRCUIT**" au format **csv** (utiliser le **Point-virgule** comme séparateur de champs, le **point** comme séparateur décimal, 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 "**formula1**" dans votre dossier de travail.
- 2) Choisir les méthodes et les propriétés adéquates, à partir de l'annexe de la page **3**, pour écrire dans le fichier créé les scripts permettant de/d' :
 - a) importer le contenu du fichier "**circuit.csv**" et le stocker dans un DataFrame à nommer "**circ**",
 - b) afficher les informations relatives au DataFrame "**circ**",
 - c) renommer la colonne "**NbTours**" par "**NombreDeTours**",
 - d) Ajouter 2000 mètres aux longueurs de tous les circuits,
 - e) calculer, dans une nouvelle colonne à nommer "**DistanceTotale**", la distance totale à parcourir puis afficher le DataFrame "**circ**". Sachant que : **DistanceTotale = LongCir * NombreTours**,
 - f) créer un graphique en barres vertes, à partir du DataFrame "**circ**", représentant, pour chaque nom de circuit, la longueur correspondante. Attribuer le titre "**Longueur circuit**" pour le graphique.

Grille d'évaluation	Partie	A (12 points)					B (8 points)						
	Question	1	2	3	4	5	1	2.a.	2.b.	2.c.	2.d.	2.e.	2.f.
	Note	0.5	4.5	2	4	1	0.5	1.25	0.5	1.25	1.25	1.25	2

ANNEXE

Les méthodes et les propriétés à utiliser avec la bibliothèque pandas

Catégorie	Syntaxe
Création d'un DataFrame	<code>IdDataFrame = pandas.DataFrame ({"NomColonne1":["Val1",..., "ValN"],..., "NomColonneM":["Val1, ..., ValN"]})</code>
Importation des données dans un DataFrame	<code>IdDataFrame = pandas.read_excel ("Chemin/ Nom_Fichier.extension", "Nom_Feuille")</code> <code>IdDataFrame = pandas.read_csv ("Chemin/Nom_Fichier.extension", sep = "séparateur")</code>
Manipulation d'un DataFrame	<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.describe ()</code> ou bien <code>print (IdDataFrame.describe ())</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>
Affichage des données d'un DataFrame	<code>IdDataFrame</code> ou bien <code>print (IdDataFrame)</code> <code>IdDataFrame ["NomColonne"]</code> ou bien <code>print (IdDataFrame ["NomColonne"])</code> <code>Id_DataFrame [["NomColonne1", "NomColonne2", ...]]</code> ou bien <code>print (Id_DataFrame [["NomColonne1", "NomColonne2", ...]])</code> <code>IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame ["NomColonne"] [N° LigneInitial : N° LigneFinal])</code> <code>IdDataFrame.iloc [N° Ligne]</code> ou bien <code>print (IdDataFrame.iloc [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.iloc [N° LigneInitial : N° LigneFinal]</code> ou bien <code>print (IdDataFrame.iloc [N° LigneInitial : N° LigneFinal])</code> <code>IdDataFrame.iloc [N° Ligne , N° Colonne]</code> ou bien <code>print (IdDataFrame.iloc [N° Ligne , N° Colonne])</code>
Modification et ajout des données dans un DataFrame	<code>IdDataFrame.loc [N° ligne , "NomColonne"] = Valeur (ou Formule)</code> <code>Id_DataFrame.loc [N° ligne] = [Liste_Valeur]</code> <code>Id_DataFrame ["NomColonne"] = Valeur (ou Formule)</code>
Nettoyage d'un DataFrame	<code>Resultat = Id_DataFrame.drop_duplicates ()</code> <code>Resultat = Id_DataFrame.dropna ()</code> <code>Resultat = Id_DtaFrame.dropna (axis = 1)</code>
Les fonctions statistiques	<code>mean ()</code> – <code>min ()</code> – <code>max ()</code> – <code>sum ()</code> – <code>count ()</code>
Affichage des données d'un DataFrame selon une ou plusieurs conditions	<code>IdDataFrame [IdDataFrame ["NomColonne"] OperateurComparaison Valeur]</code> <code>IdDataFrame [(IdDataFrame ["NomColonne"] OperateurComp Valeur) OperateurLogique (IdDataFrame ["NomColonne"] OperateurComp Valeur)]</code>
Tri des éléments d'un DataFrame	<code>DataFrame_Trié = IdDataFrame.sort_values (by = ["NomColonne1" , "NomColonne2" , ...] , ascending = [True/False , True/False , ...])</code>
Création d'un graphique	<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>