Exercice 1

Écrire un script interactif affichant un menu en boucle (tant que l'option 5 n'est pas choisie). Le script doit être robuste (validation des entrées, gestion d'erreurs) et journaliser chaque action dans un fichier diagnostique.log (date/heure + action + résultat).

Contraintes

- Le menu apparaît après chaque opération, jusqu'à ce que l'utilisateur choisisse 5.
- Utiliser au moins une boucle for ou foreach et des if pour vos sélections (n'utilisez pas uniquement Where-Object pour filtrer).
- Validation stricte des saisies (entiers positifs, chemins existants).
- Aucune opération destructive sans confirmation O/N.
- Tous les messages affichés à l'utilisateur sont en français (sans erreurs d'orthographe).
- Journaliser dans **diagnostique.log**: [AAAA-MM-JJ HH:MM:SS] <Action> ------ <Statut>. Avec Action est l'opération réalisée et Statut est l'état de la réalisation (**OK** si l'opération s'est bien déroulée ou message d'erreur si l'opération ne s'est pas bien déroulée)

Menu et fonctionnalités

Menu à afficher (3 points):

==== Menu Diagnostique ====

1) Top N processus (mémoire)

2) Supprimer le plus gros fichier d'un dossier

3) Déplacer .txt vers un sous-dossier

4) Compresser un dossier en ZIP

5) Quitter

Votre choix (1-5) : _

1. **Afficher les n processus les plus gourmands en mémoire (5 pts)**

- Demander un nombre n.
- Trier les processus par mémoire (WS/WorkingSet) décroissante.
- Afficher au maximum n éléments avec : PID, Nom, Mémoire (Mo).
- Cas limites : si n > nombre de processus, afficher seulement ceux disponibles.

2. **Supprimer le fichier le plus volumineux d'un dossier (5 pts)**

- Demander un chemin de dossier.
- Si le dossier est vide ou ne contient pas de fichiers, afficher un message explicite.
- Identifier le plus volumineux. En cas d'égalité, supprimer un de votre choix.
- Demander confirmation O/N avant la suppression.
- Journaliser le chemin supprimé et la taille.

3. **Déplacer tous les .txt d'un dossier vers un nouveau sous-dossier (5 pts)**

- Demander un dossier source ; vérifier l'existence.
- Créer automatiquement un sous-dossier MigrationTXT_yyyyMMdd_HHmmss à l'intérieur du dossier saisi.
- Déplacer tous les fichiers avec extension .txt.
- Si aucun .txt, l'indiquer sans erreur.
- En cas de conflit de nom, renommer en ajoutant _(1), _(2), etc.

4. **Compresser un dossier en nouveau fichier archive (5 pts)**

- Demander un chemin de dossier
- Créer une archive ZIP dans le même dossier parent. Utiliser le nom que vous voulez, mais ne jamais écraser.
- Journaliser le chemin de l'archive créée et la taille finale.

5. **Quitter (2 pts)**

- Afficher un message d'au revoir, journaliser et terminer le programme.
- Seule l'option 5 quitte l'application.

Barème (20 pts)

- Menu en boucle: 3 pts
- (1) Processus mémoire : 5 pts
- (2) Suppression plus gros fichier (+ confirmation + égalités) : 5 pts
- (3) Déplacement .txt (création dossier + conflits) : 5 pts
- (4) Compression (sans écraser + nom horodaté) : 5 pts
- (5) Quitter proprement : 2 pts

Exercice 2

Dans la trace **test3.csv** on observe des requêtes DNS vers des domaines en .tk. Ces requêtes peuvent indiquer une compromission (machines « infectées » faisant des requêtes vers des domaines malicieux). L'objectif est d'écrire un script interactif (menu) permettant d'extraire des informations utiles depuis la trace.

Le script doit répéter le menu jusqu'à ce que l'utilisateur choisisse 6 pour quitter.

Menu (chaque option = 5 pts sauf 6 = 0 pts)

1. Lister sans doublons les adresses IP source des machines infectées (machines qui font des requêtes DNS pour un domaine se terminant par .tk) (5 pts).
    - Définitions : on s'intéresse aux sources de requêtes DNS (pas aux réponses).
2. Afficher la liste des protocoles distincts présents dans la trace (sans doublons). (5 pts)
3. Pour une adresse IP saisie, afficher la liste (chronologique) des requêtes effectuées par cette adresse en tant que source.
    - Afficher au minimum : Time (sous format AA-MM-JJ :HH-MM-SS), Destination, Protocol, Info (ou colonnes disponibles).
    - Pour trouver le temps exact, je vous rappelle que la date et l'heure de référence dans ce fichier sont 24-01-01 22:00:00. Autrement dit : la première entrée du fichier a pour valeur Time = 0.000000 = 0 milliseconde, ce qui correspond à 24-01-01 22:00:00. Pour toutes les autres lignes, la valeur du champ Time est un décalage (exprimé en millisecondes) à ajouter à cette date/heure de référence. Exemple : 4.002629 est équivalent à 4002629 millisecondes (5 pts)
4. Pour un protocole saisi, afficher la liste des adresses IP sources distinctes qui ont utilisé ce protocole. (5 pts)
5. Afficher les tentatives d'ouverture de session TCP : lister toutes les lignes contenant SYN (ou Info indiquant SYN) et afficher : Source IP, Destination IP, Source Port, Destination Port, Time (On veut que les requêtes d'ouverture de session). (5 pts)
6. Quitter le programme (0 pt, mais obligatoire pour terminer).

Remarque sur les deux exercices :

- Ajouter des commentaires.
- Vous avez le droit d'utiliser des forums pour chercher de l'information.
- Toute utilisation de l'intelligence artificielle est considérée comme un plagiat.
- Si vous avez trouvé un bout de code ailleurs et vous l'avez utilisé, vous devez absolument ajouter sous forme de commentaire le lien de votre source d'information faute de quoi c'est du plagiat