

EXAMEN DU BACCALAURÉAT * * * Session de contrôle 2026RÉPUBLIQUE TUNISIENNE
MINISTÈRE DE L'ÉDUCATIONÉpreuve :
Algorithmique et ProgrammationSection :
Sciences de l'informatiqueDurée : **3h**Coefficient de l'épreuve : **2**N° d'inscription *Le sujet comporte 4 pages numérotées de 1/4 à 4/4.***Exercice 1 (4 points)**Soit l'algorithme de la fonction **Inconnue1** suivant :

Fonction Inconnue1 (Ch : Chaîne de caractères , c : Caractère) :

DEBUT

Si (Pos(Majus(c) , Majus(Ch)) = -1) Alors

Retourner Ch

Sinon

Ch ← Effacer(Ch , Pos(Majus(c) , Majus(Ch)) , Pos(Majus(c) , Majus(Ch))+1)

Retourner Inconnue1(Ch , c)

FinSi

FIN

Travail demandé :

- 1) Quel est le type du résultat de la fonction **Inconnue1** ?
- 2) Donner la trace d'exécution de la fonction **Inconnue1** pour l'appel suivant :
Inconnue1("Trottoir" , "T")
- 3) En déduire le rôle de la fonction **Inconnue1**.
- 4) Ecrire un algorithme d'une fonction récursive **Inconnue2(Ch , c , p)** qui permet de supprimer toutes les occurrences d'un caractère **c** dans une chaîne de caractères **Ch**, à partir d'une position **p**, sans distinction entre majuscules et minuscules.

Exemple :**Inconnue2("Trottoir" , "T" , 3)** retourne **"Trooir"**.

- 5) En faisant appel à la fonction **Inconnue2**, écrire un algorithme d'une fonction **Supprime(Ch)** qui, pour chaque caractère de la chaîne **Ch**, supprime toutes ses occurrences dans **Ch** sauf la première, sans distinction entre majuscules et minuscules.

Exemple : Pour **Ch = "Trottoir"** le résultat retourné est **"Troir"**.

En effet :

- Toutes les occurrences du caractère **"T"** (majuscule ou minuscule) dans **Ch**, sauf la première, seront supprimées. On obtient alors la chaîne **Ch = "Trooir"**.
- Toutes les occurrences du caractère **"r"** (majuscule ou minuscule) dans **Ch**, sauf la première, seront supprimées. On obtient alors la chaîne **Ch = "Trooi"**.
- Toutes les occurrences du caractère **"o"** (majuscule ou minuscule) dans **Ch**, sauf la première, seront supprimées. On obtient alors la chaîne **Ch = "Troir"**.
- Le caractère **"i"** (majuscule ou minuscule) se trouve une seule fois dans **Ch** ; il n'y a aucune occurrence supplémentaire à supprimer et le résultat retourné est **"Troir"**.

Exercice 2 (6 points)

Une entreprise de télécommunication prévoit d'installer plusieurs antennes dans une région afin d'améliorer la couverture réseau. Cette région est représentée par une matrice **M** de **L** lignes et **C** colonnes dont chaque case contient soit **1** soit **0** :

- **1** : La zone est couverte par le réseau.
- **0** : La zone n'est pas couverte.

On appelle zone couverte isolée toute case contenant la valeur **1** et entourée sur ses quatre côtés (haut, bas, gauche et droite) uniquement par des cases contenant la valeur **0**. Les quatre cases situées aux coins diagonaux d'une case (haut-gauche, haut-droite, bas-gauche et bas-droite) ne sont pas prises en compte pour déterminer si cette case est couverte isolée.

Pour les cases situées sur les bords de la matrice, tout côté inexistant est considéré comme une case contenant la valeur **0**.

L'absence de liaison entre ces zones couvertes isolées et le reste du réseau pose problèmes. L'objectif est de les identifier afin d'optimiser le plan de couverture.

Exemple :

Pour la matrice **M** suivante :

| | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 | 0 |
| 2 | 1 | 0 | 0 | 1 | 0 |
| 3 | 0 | 0 | 0 | 0 | 1 |
| 4 | 0 | 1 | 0 | 1 | 0 |
| 5 | 1 | 0 | 0 | 1 | 0 |

Les emplacements des zones couvertes isolées sont : **(0, 0) (0, 4) (2, 0) (2, 3) (3, 4) (4, 1) (5, 0)**

Le nombre de ces zones est **7**.

Travail demandé :

- 1) Déclarer un type **Mat** pour la matrice **M** de **L** lignes et **C** colonnes (avec $2 \leq L \leq 100$ et $2 \leq C \leq 100$).
- 2) Ecrire un algorithme d'une procédure **Saisie(M, L, C)** qui permet de saisir **L** et **C**, puis de remplir la matrice **M**, tout en respectant les contraintes décrites précédemment.
- 3) Ecrire un algorithme d'une procédure **ZonesIsolees(M, L, C)** qui permet :
 - d'afficher les emplacements des zones couvertes isolées dans la matrice **M**,
 - d'afficher le nombre total des zones couvertes isolées,
 - de remplir un fichier texte **F** avec les valeurs de la matrice **M**, de sorte que :
 - Chaque ligne du fichier **F** contient une ligne de la matrice **M**.
 - Les valeurs de chaque ligne de la matrice **M** soient écrites de gauche à droite dans le fichier **F**, avec un espace après chaque valeur.
 - La valeur **1** de chaque zone couverte isolée soit remplacée par le caractère "**X**".

NB : La procédure doit assurer l'ouverture et la fermeture du fichier texte situé sur la racine du disque **D** et dont le nom physique est "**Isolees.txt**".

Exemple : Pour le contenu de la matrice **M** précédent, le contenu du fichier **F** est :

| | | | | |
|---|---|---|---|---|
| X | 0 | 0 | 0 | X |
| 0 | 1 | 1 | 0 | 0 |
| X | 0 | 0 | X | 0 |
| 0 | 0 | 0 | 0 | X |
| 0 | X | 0 | 1 | 0 |
| X | 0 | 0 | 1 | 0 |

Exercice 3 (10 points)

Dans le but d'améliorer la qualité de ses produits, une entreprise souhaite réaliser une enquête afin de collecter et d'analyser les appréciations de ses adhérents. Le nombre total d'adhérents est inférieur ou égal à **1000**.

Les informations d'accès des adhérents (login et mot de passe) sont stockées dans un fichier texte nommé **"LoginPassword.txt"**. Les mots de passe sont cryptés afin d'assurer la confidentialité des comptes.

Chaque ligne du fichier contient :

- un login constitué exactement de 8 chiffres
- suivi du caractère #
- suivi d'un mot de passe d'au moins 4 caractères crypté sous forme d'une séquence binaire où chaque caractère est représenté sur un octet (8 bits).

Exemple : 65458798#00100100111100111011011001000011

← Login
← Mot de passe crypté →

Pour exploiter ce fichier, on doit procéder au décryptage des mots de passe.

Le décryptage d'un mot de passe se fait comme suit :

Pour chaque octet du mot de passe :

- Etape 1 : Convertir l'octet de la base **2** vers la base **16**.
- Etape 2 : Permuter les deux chiffres du nombre obtenu dans l'étape 1.
- Etape 3 : Convertir le nombre hexadécimal obtenu dans l'étape 2 vers la base **10**.
- Etape 4 : Déterminer le caractère dont le code ASCII est le nombre obtenu dans l'étape 3.

Exemple :

Le mot de passe crypté est : 00100100 11110011 10110110 01000011

| | | | | |
|-------------------------|-----|-----|-----|-----|
| Résultat de l'étape 1 : | 24 | F3 | B6 | 43 |
| | ↓ | ↓ | ↓ | ↓ |
| Résultat de l'étape 2 : | 42 | 3F | 6B | 34 |
| | ↓ | ↓ | ↓ | ↓ |
| Résultat de l'étape 3 : | 66 | 63 | 107 | 52 |
| | ↓ | ↓ | ↓ | ↓ |
| Résultat de l'étape 4 : | "B" | "?" | "k" | "4" |

D'où le mot de passe décrypté est : **"B?k4"**

Travail demandé :

- 1) Ecrire un algorithme d'une procédure **Decryptage(TLP , NE)** qui permet de remplir un tableau d'enregistrements **TLP**, à partir du fichier **"LoginPassword.txt"**, par les logins des adhérents ainsi que leurs mots de passe décryptés comme décrit précédemment. Chaque enregistrement du tableau **TLP** est composé des deux champs suivants :
 - **Login** : le login d'un adhérent.
 - **Pwd** : le mot de passe de l'adhérent décrypté selon le principe décrit précédemment.

NB :

- **NE** représente le nombre d'enregistrements dans le tableau **TLP**.
- Le fichier "**LoginPassword.txt**" est déjà rempli et enregistré dans le dossier "**D:\Analyse**".

2) Ecrire un algorithme d'une procédure **Remplir(TLP , NE)** qui saisit le nombre **N** de participants à l'enquête (avec $10 \leq N \leq NE$), puis permet pour chaque participant :

a) L'authentification :

- Saisir le login et le mot de passe non cryptés.
- Vérifier que ce couple (login, mot de passe) existe dans le tableau **TLP**.
 - Si le couple n'existe pas, afficher le message "**Vous n'êtes pas un adhérent**",
 - Sinon, passer à l'étape **b**).

b) La saisie de l'appréciation :

- Si l'adhérent n'existe pas dans le fichier "**Notes.dat**", lui demander de saisir une note d'appréciation de l'intervalle [1..5] et ajouter ensuite un nouvel enregistrement dans le fichier "**Notes.dat**". Chaque enregistrement du fichier "**Notes.dat**" est composé des deux champs suivants :
 - **Login** : le login de l'adhérent.
 - **Note** : la note attribuée par l'adhérent.
- Si l'adhérent existe déjà dans le fichier "**Notes.dat**", afficher un message indiquant qu'il a déjà attribué une note.

NB :

- Le fichier "**Notes.dat**" est déjà créé dans le dossier "**D:\Analyse**".
- On pourra utiliser la fonction **Chercher(Login)**, qui retourne **Vrai** si un enregistrement du fichier "**Notes.dat**" contient le login de l'adhérent (**Login**) et retourne **Faux** dans le cas contraire, que le candidat n'est pas appelé à la développer.

3) Ecrire un algorithme d'une procédure **Statistique(Moy , P)** qui permet de calculer :

- La moyenne des notes **Moy**.
- Le pourcentage **P** des notes supérieures ou égales à **4**, sachant que :

$$P = (\text{nombre de notes} \geq 4) * 100 / N$$

4) Ecrire un algorithme du programme principal qui fait appel aux trois procédures **Decryptage**, **Remplir** et **Statistique** pour collecter les appréciations des adhérents et afficher la moyenne des notes et le pourcentage des notes supérieures ou égales à **4**.