COLLEGE PRIVE LAÏC MONGO BETIB.P 972 TEL. /22 68 62 97 33 20 67 23-YAOUNDE					
ANNÉE SCOLAIRE	**SÉQUENCE**	**EPREUVE**	**CLASSE**	**DUREE**	**COEFFICIENT**
2025/2026	N°3	INFORMATIQUE	TlesC,D	02H	02
Nom du professeur : BIEM KIT MAKONGO EMMANUEL					
NOMS ET PRENOMS:					

I. ALGORITHMIQUE ET PROGRAMMATION (12 POINTS)

EXERCICE1 : ALGORITHMIQUE (8pts)

Un élève est caractérisé par son Nom, son Age et sa Moyenne.

1) Créer un nouveau type **Elève** ayant les caractéristiques ci-dessus : **2pts**

2) Déclarer un tableau nommé **E1** de 50 élèves **1pt**

3) Citer deux (02) éléments nécessaires pour effectuer une recherche séquentielle dans un tableau d'Elèves : ---------- **1pt**

4) Donner un avantage de la recherche séquentielle sur la recherche dichotomique : ----------

---

---------- **1pt**

5) Ecrire un algorithme qui recherche un élève à partir de son nom et affiche son enregistrement (Nom, Age, Moyenne) s'il est trouvé **3pts**

http://sujetexa.com

EXERCICE2 : PROGRAMMATION EN C (4pts)

On considère que les tableaux **jours** et **annees** représentés avec les données suivantes :

annees	2022	2023	2024	2025	2026

jours	"Lundi"	"Mardi"	"Mercredi"	"Jeudi"	"Vendredi"	"Samedi"	"Dimanche"

1) Déclarer et initialiser en C le tableau ***annees.*** **1pt**

2) Ecrire l'instruction en C ayant permis d'affecter l'année ***2025*** à la position appropriée du tableau. **1pt**

3) c) Traduire en C l'instruction suivante puis donner son résultat : Ecrire (annees[3], jours [4]) : **2pts**

II. SYSTÈME D'INFORMATION ET BASE DE DONNEES (8 POINTS)

1) Définir **Requête :** ----------
---------- **1pt**

2) Donner la différence entre

a) Un **Identifiant** et une **Clé Primaire** : ----------
---------- **1pt**

b) **ALTER** et **UPDATE :** ----------
---------- **1pt**

3) On considère la table suivante : **COMMANDE (NumCOMM, #NumSIRET, #NumPROD, QUANTITÉ)**

a) Ecrire la requête SQL qui permet de créer cette table sachant que tous les champs ont le même type de données (Entier).

2pts

b) Ecrire la requête qui permet de supprimer le champ QUANTITE de cette table **1pt**

c) Expliquer ce que fait la requête suivante : ***ALTER TABLE COMMANDE RENAME TO FACTURE ;*** ----------
---------- **1pt**

Joyeux Noël et Bonne Année 2026

Correction de l'épreuve d'informatique

I. ALGORITHMIQUE ET PROGRAMMATION

EXERCICE 1 : ALGORITHMIQUE (8 pts)

1) Créer un nouveau type `Elève` (2 pts)

```
Type Eleve = Enregistrement
    Nom : chaine de caracteres
    Age : entier
    Moyenne : reel
Fin
```

2) Déclarer un tableau `E1` de 50 élèves (1 pt)

```
Var E1 : Tableau[1..50] de Eleve
```

3) Deux éléments nécessaires pour une recherche séquentielle dans un tableau d'élèves (1 pt)

Il faut :

— le tableau dans lequel on effectue la recherche, ici `E1` ;

— la valeur recherchée, ici le nom de l'élève.

On peut aussi citer la taille du tableau, c'est-à-dire 50.

4) Avantage de la recherche séquentielle sur la recherche dichotomique (1 pt)

La recherche séquentielle ne nécessite pas que le tableau soit trié. Elle est donc utilisable directement sur un tableau quelconque, contrairement à la recherche dichotomique qui exige un tableau trié.

5) Algorithme de recherche d'un élève à partir de son nom (3 pts)

```
Algorithme Recherche_Eleve
Var
    i : entier
    nomRech : chaine de caracteres
    trouve : booleen
Debut
    Ecrire("Entrer le nom de l'eleve a rechercher : ")
    Lire(nomRech)

    i <- 1
    trouve <- faux

    TantQue (i <= 50) ET (trouve = faux) Faire
        Si E1[i].Nom = nomRech Alors
            trouve <- vrai
        Sinon
            i <- i + 1
```

```
        FinSi
    FinTantQue

    Si trouve = vrai Alors
        Ecrire("Nom : ", E1[i].Nom)
        Ecrire("Age : ", E1[i].Age)
        Ecrire("Moyenne : ", E1[i].Moyenne)
    Sinon
        Ecrire("Eleve non trouve")
    FinSi
Fin
```

EXERCICE 2 : PROGRAMMATION EN C (4 pts)

1) Déclarer et initialiser en C le tableau `annees` (1 pt)

```
int annees[5] = {2022, 2023, 2024, 2025, 2026};
```

2) Instruction en C pour affecter l'année 2025 à la position appropriée (1 pt)

L'année 2025 se trouve à l'indice 3 :

```
annees[3] = 2025;
```

3) Traduire en C l'instruction `Ecrire(annees[3], jours[4])` puis donner son résultat (2 pts)

Traduction en C :

```
printf("%d %s\n", annees[3], jours[4]);
```

Résultat :

```
2025 Vendredi
```

Car :
- `annees[3] = 2025;`
- `jours[4] = "Vendredi"`.

II. SYSTÈME D'INFORMATION ET BASE DE DONNÉES

1) Définir une requête (1 pt)

Une requête est une instruction adressée à un système de gestion de base de données, généralement en langage SQL, permettant de rechercher, d'extraire, d'insérer, de modifier ou de supprimer des données dans une base de données.

2) Différences (2 pts)

a) Identifiant et clé primaire (1 pt) Un identifiant est un attribut ou un ensemble d'attributs permettant d'identifier de manière unique un enregistrement d'une table.

La clé primaire est l'identifiant choisi et déclaré comme clé principale de la table. Elle impose l'unicité et l'interdiction des valeurs nulles.

b) `ALTER` et `UPDATE` (1 pt)

- `ALTER` est une commande de définition de données : elle modifie la structure d'une table, par exemple ajouter, supprimer ou renommer une colonne.
- `UPDATE` est une commande de manipulation de données : elle modifie les valeurs des enregistrements existants dans une table.

3) Table `COMMANDE (NumCOMM, #NumSIRET, #NumPROD, QUANTITÉ)`

a) Requête SQL de création de la table (2 pts) Tous les champs sont de type entier :

```
CREATE TABLE COMMANDE (
    NumCOMM INT,
    NumSIRET INT,
    NumPROD INT,
    QUANTITE INT
);
```

On peut aussi préciser `NumCOMM` comme clé primaire :

```
CREATE TABLE COMMANDE (
    NumCOMM INT PRIMARY KEY,
    NumSIRET INT,
    NumPROD INT,
    QUANTITE INT
);
```

Les symboles # indiquent généralement des clés étrangères, mais les tables référencées ne sont pas données dans l'énoncé.

b) Requête SQL pour supprimer le champ `QUANTITÉ` (1 pt)

```
ALTER TABLE COMMANDE DROP COLUMN QUANTITE;
```

c) Expliquer la requête suivante (1 pt)

```
ALTER TABLE COMMANDE RENAME TO FACTURE;
```

Cette requête renomme la table `COMMANDE` en `FACTURE`. La structure et les données de la table sont conservées ; seul le nom de la table change.