Centre universitaire de Mila

Institut des Sciences et de la Technologie

Département de Mathématique et Informatique

1ère Année LMD MI

**Série de Td N°1**

Module : Algorithmique et structures de données 2 Année universitaire : 2018/2019

**Exercice 1**

Une clinique veut automatiser la gestion de ses patients (malades) et vous confie cette tache. Pour cela, un patient sera identifié par : numéro de patient, nom, prénom, date de naissance (Jour, Mois et Année), numéro de téléphone, adresse et le nom de sa maladie.

1. Écrire les structures de données nécessaires à la définition d’un patient et d’une structure permettant la gestion de 1800 patients ?
2. Ecrire un algorithme permettant d’afficher un patient recherché par son numéro ?
3. Modifier l’algorithme pour qu’il affiche tous les patients de la wilaya de Mila ?

**Exercice 2**

La Wilaya de Mila veut automatiser la gestion de ses communes et vous confie cette tache. Pour cela, une commune sera identifié par : numéro de la commune, nom, date de création (Jour, Mois et Année), surface et nombre d’habitants.

1) Écrire les structures de données nécessaires à la définition d’une commune et d’une structure permettant la gestion de 50 communes ?

2) Écrire un algorithme qui réalise les tâches suivantes:

a) Saisir ***n*** communes dans le tableau.

b) Afficher les communes qui ont un nombre d’habitants inferieur à un nombre ***Nbr*** et leurs dates de création **ne dépasse pas 15 ans** ?

**Exercice 3**

L’administration du département MI veut automatiser la gestion de ses étudiants et vous confie cette tache. Pour cela, un étudiant sera identifié par :

son numéro d’étudiant, son nom, son prénom, sa date de naissance (qui sera décomposée en jour mois et année de type entier), année d’étude, (entier entre 1 et 5), un tableau contenant les moyennes aux 6 UE préparées (de l’année en cours).

1. Écrire en algorithmique les structures de données nécessaires à la définition d’un étudiant et d’une structure permettant la gestion de 1500 étudiants.
2. Écrire un algorithmique permettant les tâches suivantes:

a) Rechercher un étudiant par son numéro.

b) Afficher des étudiants recherchés par nom et prénom.

**Solution**

**Exercice 1**

1) déclaration des structures nécessaires

**Type**

Structure date

 jour : entier;

 mois : entier;

 année: entier;

FinStructure ;

Structure patient

 num : entier;

 nom: chaine;

 prénom: chaine;

 date\_nais: date;

 tél : entier;

 adresse: chaine;

 maladie: chaine;

FinStructure ;

TP [1800]: tableau de patient

2) afficher un patient recherché par son numéro ?

Début

Ecrire (‘Donnez le numéro du patient:’ ) ;

Lire (N); // N est une variable de type entier

Pour i allant de 1 à 1800 faire

Si (TP[i].Num= N ) alors

Ecrire (TP[i].Nom);

Ecrire (TP[i].prénom);

Finsi

Fin pour

Fin .

3) afficher tous les patients de la wilaya de Mila ?

Problème : l’adresse du patient est une chaine de caractères, donc on ne peut pas extraire la wilaya de cette forme.

Solution : on doit déclarer le champ adresse sous forme de structure qui contient un champ wilaya de type chaine de caractères comme suit :

Structure adr

 wilaya : chaine;

 commune :chaine ;

 Num\_maison: entier;

FinStructure ;

Dans ce cas, la structure patient devient comme suit :

Structure patient

 num : entier;

 nom: chaine;

 prénom: chaine;

 date\_nais: date;

 tél : entier;

 adresse: adr;

 maladie: chaine;

FinStructure ;

Et l’algorithme est le suivant :

Début

Pour i allant de 1 à 1800 faire

Si (**TP[i].Adresse.wilaya** = ’’Mila’’ ) alors

Ecrire (TP[i].Nom);

Ecrire (TP[i].prénom);

Finsi

Fin pour

Fin .

**Exercice 2**

**1) les déclarations nécessaires**

**Type**

Structure date

 jour : entier;

 mois : entier;

 année: entier;

FinStructure ;

Structure commune

Num : entier ;

Nom : chaine ;

Date\_creation : **date**;

Surface : réel ;

Nbr\_habit : entier ;

Finstructure

C[50] : tableau de commune ;

2) l’algorithme :

Algorithme gestion\_communes

**// ici en recopie toute la déclaration précédente**

i,n, nbr : entier ; // **n** pour les communes à saisir , et **nbr** pour le nombre d’habitant recherché

Début

Ecrire (‘donnez le nombre de communes à insérer’) ;

Lire (n) ;

**Pour** i allant de 1 à n faire

Ecrire (‘donnez le numéro de la commune :’) ;

Lire (C[i].num) ;

Ecrire (‘donnez le nom de la commune :’) ;

Lire (C[i].nom) ;

Ecrire (‘donnez le jour de création de la commune :’) ;

Lire (C[i].Date\_creation.jour) ;

Ecrire (‘donnez le mois de création de la commune :’) ;

Lire (C[i].Date\_creation.mois) ;

Ecrire (‘donnez l’année de création de la commune :’) ;

Lire (C[i].Date\_creation.année) ;

Ecrire (‘donnez la surface de la commune :’) ;

Lire (C[i].surface) ;

Ecrire (‘donnez le nombre d’habitant de la commune :’) ;

Lire (C[i].Nbr\_habit) ;

**finpour**

//b) Afficher les communes qui ont un nombre d’habitants inferieur à un nombre ***Nbr*** et leurs dates //de création **ne dépasse pas 15 ans** ?

Ecrire (‘donnez le nombre d’habitant recherché’) ;

Lire (Nbr) ;

Pour i allant de 1 à n faire

 Si (C[i].Nbr\_habit< Nbr) et (C[i]. Date\_creation.année+15<2019) faire

 Ecrire (C[i].nom) ;

 Finsi

Finpour

Fin.