

## Algorithmique II

### -TD 3- Enregistrements/Fichiers

#### Exercice 1

Écrire un algorithme considérant un enregistrement Étudiant, caractérisé par les données suivantes : nom, prénom, sexe et âge. Ledit algorithme permet de

#### Algorithme info\_étudiants\_1

- Créer la structure Enregistrement

Etudiant : Enregistrement

nom : chaîne de caractères

prenom : chaîne de caractères

sexe : caractère

age : entier

fenreg

- Remplir les éléments d'un premier enregistrement étudiante par des données saisies au clavier

#### Variables

e1 : Etudiant

e2 : Etudiant

#### début

e1.sexe ← 'F'

écrire("Saisissez le nom de l'étudiant(e) : ")

lire(e1.nom)

écrire("Saisissez le prénom de l'étudiant(e) : ")

lire(e1.prenom)

écrire("Saisissez l'âge de l'étudiant(e) : ")

lire(e1.age)

- Afficher ses informations à l'écran

si e1.sexe = 'M' alors

écrire("Bonjour M. ", e1.prenom+" "+e1.nom, " Vous avez ", e1.age, " ans.")

sinon

écrire("Bonjour Mlle ", e1.prenom+" "+e1.nom, " Vous avez ", e1.age, " ans.")

fsi

- Vérifier si l'étudiante est majeure/mineure et afficher le résultat

si e1.age >= 18 alors

écrire("Vous êtes Majeure.")

sinon

écrire("Vous êtes Mineure.")

fsi

- Ajouter l'étudiante suivante qui porte le même nom et prénom (avec moins d'instructions)

e2.nom ← e1.nom

e2.prenom ← e1.prenom

e2.sexe ← "F"

écrire("Saisissez l'âge de l'étudiant(e) : ")

lire(e2.age)

```
- Afficher l'ordre (première ou deuxième) de l'étudiante la plus jeune
si e1.age > e2.age alors
    écrire("Le(a) deuxième étudiant(e) est plus jeune.")
sinon
    écrire("Le(a) premier(ère) étudiant(e) est plus jeune.")
fsi
fin
```

## Exercice 2

Écrire un algorithme considérant un enregistrement Étudiant, caractérisé par les données suivantes : nom, prénom, sexe et date de naissance (jour, mois, année). Ledit algorithme permet de

### Algorithme info\_étudiants\_2

```
- Créer les structures Enregistrement nécessaires
Date_Naiss : Enregistrement {on doit commencer par l'enregistrement Date_Naiss car il va être
    jour : entier                utilisé par la suite dans la déclaration de Etudiant}
    mois : entier
    annee : entier
```

```
fenreg
Etudiant : Enregistrement
    nom : chaîne de caractères
    prenom : chaîne de caractères
    sexe : caractère
    d_naiss : Date_Naiss
fenreg
```

```
- Remplir les éléments d'un premier enregistrement étudiant par des données saisies au clavier
```

### Variables

```
e1 : Etudiant
e2 : Etudiant
naiss1 : entier
naiss2 : entier
```

### début

```
écrire("Saisissez le nom de l'étudiant(e) : ")
lire(e1.nom)
écrire("Saisissez le prénom de l'étudiant(e) : ")
lire(e1.prenom)
écrire("Saisissez le sexe de l'étudiant(e) : ")
lire(e1.sexe)
écrire("Saisissez la date de naissance de l'étudiant(e) ")
écrire("Jour : ")
lire(e1.d_naiss.jour)
écrire("Mois : ")
lire(e1.d_naiss.mois)
écrire("Année : ")
lire(e1.d_naiss.annee)
```

```

- Afficher ses informations à l'écran
si e1.sexe = 'M' alors
    écrire("Bonjour M. ", e1.prenom+" "+e1.nom, " Vous êtes né le ",
        e1.d_naiss.jour, "/", e1.d_naiss.mois, "/", e1.d_naiss.annee)
sinon
    écrire("Bonjour Mlle ", e1.prenom+" "+e1.nom, " Vous êtes née le ",
        e1.d_naiss.jour, "/", e1.d_naiss.mois, "/", e1.d_naiss.annee)
fsi

- Vérifier si l'étudiant est majeur/mineur (comparer l'année de naissance) et afficher le résultat
si (2025 - e1.d_naiss.annee) >= 18 alors
    écrire("Vous êtes Majeur.")
sinon
    écrire("Vous êtes Mineur.")
fsi

- Ajouter l'étudiant suivant qui porte la même date de naissance (avec moins d'instructions)
e2.d_naiss ← e1.d_naiss
écrire("Saisissez le nom de l'étudiant(e) : ")
lire(e2.nom)
écrire("Saisissez le prénom de l'étudiant(e) : ")
lire(e2.prenom)
écrire("Saisissez le sexe de l'étudiant(e) : ")
lire(e2.sexe)

- Afficher le nom complet de l'étudiant le plus jeune (en cas de deux étudiants de dates de
naissance différentes).
{Pour pouvoir comparer les deux dates on doit les convertir en entier, le plus simple est de
remplacer une date "12/09/2024" par exemple par l'entier 20240912}
naiss1 ← e1.d_naiss.annee * 10000 + e1.d_naiss.mois * 100 + e1.d_naiss.jour
naiss2 ← e2.d_naiss.annee * 10000 + e2.d_naiss.mois * 100 + e2.d_naiss.jour
si naiss2 > naiss1 alors
    écrire("e1.prenom+" "+e1.nom, " est plus jeune.")
sinon
    écrire("e2.prenom+" "+e2.nom, " est plus jeune.")
fsi
fin

```

### Exercice 3

Un Étudiant est caractérisé par les données suivantes : code, nom, prénom, sexe et modules. Ce dernier peut suivre un ou plusieurs modules simultanément et chaque module est caractérisé par intitulé, enseignant et note. Le nombre de module maximum est 10. Écrire un algorithme qui permet de

Algorithme info\_étudiants\_3

- Créer les structures Enregistrement nécessaires
- Module : Enregistrement
- intitulé : chaîne de caractères
- enseignant : chaîne de caractères

note : réel

fenreg

Etudiant : Enregistrement

code : entier

nom : chaîne de caractères

prenom : chaîne de caractères

sexe : caractère

modules : Tableau(10) Module

fenreg

- Remplir les éléments d'un premier enregistrement étudiant par des données saisies au clavier

Variables

e1 : Etudiant

e2 : Etudiant

note\_val : réel

N, i : entier

début

```

écrire("Saisissez le nom de l'étudiant(e) : ")
lire(e1.code)
écrire("Saisissez le nom de l'étudiant(e) : ")
lire(e1.nom)
écrire("Saisissez le prénom de l'étudiant(e) : ")
lire(e1.prenom)
écrire("Saisissez le sexe de l'étudiant(e) : ")
lire(e1.sexe)
écrire("Saisissez le nombre de modules que vous suivez : ")
lire(N)
écrire("Saisissez les infos de chaque module ")
pour i ← 0 à N-1 faire
    écrire("Saisissez les infos du module n° ", i)
    écrire("Intitulé : ")
    lire(e1.modules(i).intitule)
    écrire("Nom de l'enseignant : ")
    lire(e1.modules(i).enseignant)
    écrire("Note du module : ")
    lire(e1.modules(i).note)
fpour

```

- Demander à l'utilisateur la note de validation du module

```

écrire("Saissez le nombre de module que vous suivez : ")
lire(note_val)

```

- Afficher les intitulés des modules validés

```

écrire("Les modules validés sont : ")
pour i ← 0 à N-1 faire
    si e1.modules(i).note >= note_val alors
        écrire(e1.modules(i).intitule)
    fsi
fpour

```

#### Exercice 4

Considérons les données de l'exercice 3

Écrire une fonction qui prend en paramètre un étudiant et qui retourne la liste des modules à rattraper, le rattrapage est obligatoire si l'étudiant n'a pas une moyenne de 10/ 20 après compensation.

Fonction rattrapage (e : Etudiant) : Tableau(10) Module

Variables

\_modules : Tableau(10) Module

\_modulesVide : Tableau(0) Module

i : entier

j : entier  $\leftarrow$  1

s : réel  $\leftarrow$  0

début

pour i  $\leftarrow$  0 à N-1 faire

s  $\leftarrow$  s + e.module(i).note

si e.module(i).note < 10 alors

\_modules(j)  $\leftarrow$  e.modules(i)

j  $\leftarrow$  j+1

fsi

fpour

si (s/N)  $\geq$  10 alors {si l'étudiant a une note  $\geq$  10, on retourne un tableau vide}

Retourner ( \_modulesVide )

fsi

Retourner ( \_modules )

fin

#### Exercice 5

Considérons les données de l'exercice 3

Afin de gérer les informations des étudiants de la FSR, on dispose d'une base de données (bd) remplie desdites informations. Écrire une fonction inscrire() permettant d'ajouter un nouvel étudiant à partir des données saisies au clavier. La fonction renvoie la bd après mise à jour.

Fonction inscrire (bd : Tableau Etudiant, taille : entier) : Tableau Etudiant

Variables

e : Etudiant

N, i : entier

\_bd : Tableau(taille+1) Etudiant

début

écrire("Saisissez le code de l'étudiant(e) : ")

lire(e.code)

écrire("Saisissez le nom de l'étudiant(e) : ")

lire(e.nom)

écrire("Saisissez le prénom de l'étudiant(e) : ")

lire(e.prenom)

écrire("Saisissez le sexe de l'étudiant(e) : ")

lire(e.sexe)

écrire("Saisissez le nombre de module que vous suivez : ")

lire(N)

écrire("Saisissez les infos de chaque module ")

```

pour i ← 0 à N-1 faire
    écrire("Saisissez les infos du module n° ", i)
    écrire("Intitulé : ")
    lire(e.modules(i).intitule)
    écrire("Nom de l'enseignant : ")
    lire(e.modules(i).enseignant)
    écrire("Note moyenne : ")
    lire(e.modules(i).note)
fpour
_bd(taille) ← e
Retourner ( _bd )
fin

```

### Exercice 6

Écrire une fonction permettant de calculer le nombre d'occurrences d'un caractère donné, noté *c*, dans un fichier texte *F*. Le caractère recherché ainsi que le fichier sont donnés en paramètres.

Fonction nbOccurences(nom\_fichier : chaîne de caractères, char : caractère)

Variables

cpt : entier ← 0

c : caractère

F : FICHER

début

F ← ouvrir(nom\_fichier, "lecture")

Tant que NON( FDF(F) ) faire

lire(F, c)

si char = c alors

cpt ← cpt+1

fsi

ftq

fermer( F )

Retourner ( cpt )

fin

### Exercice 7

Écrivez un algorithme permettant de créer un fichier nommé "nombres.dat" contenant des nombres entiers saisis au clavier. Écrivez les fonctions réalisant les actions suivantes :

- Afficher à l'écran les nombres pairs du fichier.
- Déterminer le nombre d'éléments du fichier.
- Supprimer les éléments du fichier se trouvant après une position donnée.

### Exercice 8

Considérons les données de l'exercice 3

- Écrire une fonction permettant de saisir un groupe d'étudiant dans un fichier texte.
- Écrire une fonction qui permet de calculer la moyenne générale d'un groupe d'étudiants contenus dans le fichier.