

Informatique

Chloé Cabot

octobre 2020

CPII ESIGELEC 2A

Rappels

Les chaînes de caractères

- Un caractère est codé sur 1 octet en encodage ASCII, de 0 à 255 (ex : 'a' = 97)
- Une chaîne de caractères est un **tableau** de caractères
- Le caractère marquant la fin de la chaîne est le caractère spécial : '\0'
- Plusieurs opérations sur les chaînes de caractères, prédéfinies comme fonctions en C dont :
 - **afficher**(chaîne)
 - chaîne ← **saisir**(chaîne)
 - nombre ← **longueur**(chaîne)
 - chaîne_concat ← **concatener**(chaîne_1, chaîne_2, chaîne_concat)
 - nombre ← **comparer**(chaîne_1, chaîne_2)

Les enregistrements

Les types vus jusqu'à présent permettent de facilement représenter des données élémentaires ou un ensemble de ces données. Par exemple :

- une note représentée par un nombre réel,
- un ensemble de notes représenté par un tableau de réels.

Lorsqu'il s'agit de représenter des concepts composés de plusieurs données (par exemple une voiture) cela devient plus difficile à gérer. Par exemple, pour une voiture on peut avoir :

- une variable *marque* de type chaîne de caractères,
- une variable *cylindrée* de type réel,
- une variable *nb_portes* de type entier,
- etc...

C'est possible si on n'a qu'une voiture à gérer, mais devient compliqué pour un ensemble de voitures : *marque[1]*, *marque[12]*, *marque[48]*...

La solution : utiliser un **enregistrement**.

Enregistrement : définition

Les enregistrements permettent de créer de nouveaux types composés de plusieurs champs de types différents. Par exemple, une voiture sera composée des champs *marque*, *cylindrée* et *nb_portes*.


```
NomDuType : ENREGISTREMENT  
    champ1 : type1  
    champ2 : type2  
    ...  
FIN ENREGISTREMENT
```

Exemples

Voiture : ENREGISTREMENT

marque : tableau de 20 caractères

cylindree : reel

nb_portes : entier

FIN ENREGISTREMENT

Ennemi : ENREGISTREMENT

nom : tableau de 30 caractères

pt_vie : entier

arme : tableau de 30 caractères

FIN ENREGISTREMENT

Déclaration d'une variable

Une fois le nouveau type déclaré, il peut être utilisé comme n'importe quel autre type.

```
ma_variable : NomDuType
```

Par exemple pour la voiture nous avons :

```
titine : Voiture
```

On accède aux champs d'une variable de type enregistrement à l'aide de l'opérateur « . » (point)

```
# Modification des valeurs :  
titine.marque ← "Peunault"  
titine.cylindrée ← 1.2  
titine.nb_portes ← 5  
# Utilisation des valeurs  
afficher(titine.marque)  
SI titine.nb_portes > 3  
    afficher("Il y a des portes à l'arrière")  
FIN SI
```

Enregistrement imbriqué

On parle d'enregistrement imbriqué lorsque la définition d'un enregistrement contient au moins un autre enregistrement.

```
Conducteur : ENREGISTREMENT
    nom : tableau de 30 caractères
    permis : caractères
    sa_voiture : Voiture
FIN ENREGISTREMENT
```

```
Concessionnaire : ENREGISTREMENT
    adresse : tableau de 30 caractères
    catalogue : tableau de 50 Voiture
FIN ENREGISTREMENT
```

- Enregistrement = structure de données regroupant un ensemble de champs, de **types différents**

```
Ennemi : ENREGISTREMENT
    nom : tableau de 30 caractères
    pt_vie : entier
    arme : tableau de 30 caractères
FIN ENREGISTREMENT
```

- On accède à un champ avec le « . » (point)

```
gargantua : Ennemi
gargantua.pt_vie = 100
```

- On peut imbriquer des enregistrements

```
Map : ENREGISTREMENT
    dimension : entier
    ennemis : tableau de Ennemi
FIN ENREGISTREMENT
```

Exercice : Gestion de vinyles

On se propose de créer un programme de gestion de stock pour une boutique de vinyles. Lors de son entrée en stock, un vinyle est caractérisé par les données suivantes :

- Nom de l'artiste
- Nom de l'album
- Date de parution
- Style musical
- Note par les utilisateurs du site Discogs (de 1 à 5)

Question 1 : Trouver les types de données adaptés à la représentation de ces informations.

Question 1

```
Date : ENREGISTREMENT  
    jour : entier  
    mois : entier  
    annee : entier  
FIN ENREGISTREMENT
```


Question 1

Date : ENREGISTREMENT

jour : entier

mois : entier

annee : entier

FIN ENREGISTREMENT

Vinyle : ENREGISTREMENT

artiste : tableau de 50 caractères

album : tableau de 50 caractères

date_sortie : Date

style : tableau de 50 caractères

note : réel

FIN ENREGISTREMENT

Exercice : Gestion de vinyles

On se propose de créer un programme de gestion de stock pour une boutique de vinyles. Lors de son entrée en stock, un vinyle est caractérisé par les données suivantes :

- Nom de l'artiste
- Nom de l'album
- Date de parution
- Style musical
- Note par les utilisateurs du site Discogs (de 1 à 5)

Question 3 : Écrire un algorithme qui permet l'ajout d'un vinyle arrivé en stock.

Question 3

```
ALGORITHME ajout
  ENVIRONNEMENT
    ENTREES
      stock : tableau de 1000 Vinyle
      nb_vinyles : entier
    SORTIE
      néant
    INTERNES
      nvx_vinyle : Vinyle
```

Question 3

TRAITEMENT

DEBUT

ECRIRE "Artiste ? "

nvx_vinyle.artiste ← saisir(nvx_vinyle.artiste)

ECRIRE "Album? "

nvx_vinyle.album ← saisir(nvx_vinyle.album)

ECRIRE "Année parution ? "

nvx_vinyle.date_sortie.annee ← LIRE

ECRIRE "Mois parution ? "

nvx_vinyle.date_sortie.mois ← LIRE

ECRIRE "Jour parution ? "

nvx_vinyle.date_sortie.jour ← LIRE

ECRIRE "Style? "

nvx_vinyle.style ← saisir(nvx_vinyle.style)

ECRIRE "Note? "

nvx_vinyle.note ← LIRE

stock[nb_vinyles + 1] ← nvx_vinyle

FIN

FIN ALGORITHME

Exercice : Gestion de vinyles

On se propose de créer un programme de gestion de stock pour une boutique de vinyles. Lors de son entrée en stock, un vinyle est caractérisé par les données suivantes :

- Nom de l'artiste
- Nom de l'album
- Date de parution
- Style musical
- Note par les utilisateurs du site Discogs (de 1 à 5)

Question 4 : Écrire un algorithme qui liste les vinyles d'un artiste donné.

Question 4

```
ALGORITHME affiche_vinyle
  ENVIRONNEMENT
    ENTRÉES
      v : Vinyle #vinyle à afficher
    SORTIE
      néant
    INTERNES
      néant
  TRAITEMENT
    DEBUT
      afficher(v.artiste)
      afficher(v.album)
      ECRIRE v.date_sortie.jour , " / ",
v.date_sortie.mois , " / ", v.date_sortie.annee
      afficher(v.style)
      ECRIRE v.note
    FIN
  FIN ALGORITHME
```

Question 4

```
ALGORITHME liste_artiste
  ENVIRONNEMENT
    ENTRÉES
      stock : tableau de 1000 Vinyle
      nb_vinyles : entier
      artiste_recherche : tableau de 50 caractères
    SORTIE
      néant
    INTERNES
      index : entier
  TRAITEMENT
    DEBUT
      POUR index ALLANT DE 0 A nb_vinyles PAR PAS DE 1
      FAIRE
        SI compare(stock[index].artiste,
          artiste_recherche) = 0 FAIRE
          affiche_vinyle(stock[index])
        FIN SI
      FIN POUR
    FIN
  FIN ALGORITHME
```

Exercice : Gestion de vinyles

On se propose de créer un programme de gestion de stock pour une boutique de vinyles. Lors de son entrée en stock, un vinyle est caractérisé par les données suivantes :

- Nom de l'artiste
- Nom de l'album
- Date de parution
- Style musical
- Note par les utilisateurs du site Discogs (de 1 à 5)

Question 5 : Écrire un algorithme qui liste les vinyles sortis lors d'une année donnée et indiquant le nombre d'albums sortis cette année là.

Question 5

```
ALGORITHME nombre_annee
  ENVIRONNEMENT
    ENTRÉES
      stock : tableau de 1000 Vinyle
      nb_vinyles : entier
      annee_recherchee : entier
    SORTIE
      néant
    INTERNES
      index : entier
      compteur : entier
```

Question 5

```
TRAITEMENT
  DEBUT
    compteur  $\leftarrow$  0
    POUR index ALLANT DE 0 A nb_vinyles PAR PAS DE 1
  FAIRE
    SI stock[index].date_sortie.annee =
annee_recherchee FAIRE
      affiche_vinyle(stock[index])
      compteur  $\leftarrow$  compteur + 1
    FIN SI
  FIN POUR
  ECRIRE compteur , "vinyles sont sortis en " ,
annee_recherchee
FIN
FIN ALGORITHME
```

Exercice : Gestion de vinyles

On se propose de créer un programme de gestion de stock pour une boutique de vinyles. Lors de son entrée en stock, un vinyle est caractérisé par les données suivantes :

- Nom de l'artiste
- Nom de l'album
- Date de parution
- Style musical
- Note par les utilisateurs du site Discogs (de 1 à 5)

Question 6 : Écrire un algorithme qui liste les vinyles dont, pour un style donné, le score est supérieur à la moyenne de ce style (moyenne calculée sur le stock contenu en boutique).

Question 6

```
ALGORITHME liste_vinyle_sup_moy_style
  ENVIRONNEMENT
    ENTRÉES
      stock : tableau de 1000 Vinyle
      nb_vinyles : entier
      style_recherche : tableau de 50 caractères
    SORTIE
      néant
    INTERNES
      index : entier
      compteur : entier
      somme_notes : réel
      moyenne : réel
```

Question 6

TRAITEMENT

DEBUT

somme_notes \leftarrow 0

compteur \leftarrow 0

#Calcul de la moyenne du style donné

POUR index ALLANT DE 0 A nb_vinyles PAR PAS DE 1 FAIRE

SI compare(stock[index].style, style_recherche) = 0 FAIRE

somme_notes \leftarrow stock[index].note + somme_notes

compteur \leftarrow compteur + 1

FIN SI

FIN POUR

moyenne \leftarrow somme_notes / compteur

recherche les vinyles du style donné dont la note > moyenne

POUR index ALLANT DE 0 A nb_vinyles PAR PAS DE 1 FAIRE

SI compare(stock[index].style, style_recherche) = 0 ET

stock[index].note > moyenne FAIRE

affiche_vinyle(stock[index])

FIN SI

FIN POUR

FIN

FIN ALGORITHME