

# Algorithmique

## **Chap.1 : Introduction**

## **Chap.2 : Fonctions & Procédures**

Fonction Nom(args : types ) : type sortie

Procédure Nom(E/S args : types)

## **Chap.3 : Tableaux & Recherche**

Déclaration:

Type :

Tab : tableau [0..50] de types

Var :

T : tab

Redim(T[1..20]) => Pour I de 0 a 19 faire

Recherche :

Recherche(Tableau non trié):

Je parcours tout le tableau

Recherche séquentielle(tableau trié):

Je compare avec chaque élément jusqu'à trouver

l'indice voulu ou que le contenu de la case est > a

mon résultat voulu

=> je sors de la boucle

Recherche par les dichotomies(tableau trié):

```
Sup <= n-1; Inf <= 0; trouvé <= faux;
Tant que (inf<=sup) et (trouvé = faux) {
    Mid = (inf + sup) div 2
    Si (t[mid] = searched) alors
        trouvé<=vrai
    Sinon si (t[mid]<searched) alors
        inf = mid+1
    Sinon
        sup<=mid-1
    Finsi
Finsi
}
Fintantque
```

### ***Chap.4 : Algorithmes de Tri***

Tri par remplacement :

- Identifier le maximum du tableau T
- Rechercher le minimum du tableau T
- Recopier ce minimum dans Ttrié à la position i
- Remplacer le minimum du tableau T par le maximum
- Recommencer pour i+1

Tri par selection :

1	2	3	4	5	6	7
17	4	8	24	1	13	7
1	4	8	24	17	13	7
1	4	8	24	17	13	7
1	4	7	24	17	13	8
1	4	7	8	17	13	24
1	4	7	8	13	17	24
1	4	7	8	13	17	24

di Abdelmoula

i min

Tri a bulles :

On compare  $T[i]$  et  $T[i+1]$  jusqu'a plus avoir de permutations

Tri Indirect :

-Utilise un tableau auxiliaire qui indique, pour chaque élément du tableau à trier, le rang que celui-ci devrait occuper dans le tableau trié.

Principe :

-Compter, pour chaque élément du tableau à trier, le nombre d'éléments qui lui sont inférieurs ou égaux.

-Le nombre trouvé donnera la place (l'indice) de cet élément dans le tableau trié

-Le tri se fait donc ensuite par l'intermédiaire de cet index

## **Chap.5 : Enregistrements & Tableaux d'enregistrements**

## Enregistrements :

-Simples :

Type :

Nom\_type = enregistrement

Nom\_champ1 : type

Nom\_champ2 : type

Nom\_champ3 : type

Finenregistrement

-Imbriqués ( 2 Methodes ):

Type:

Nom\_type = enregistrement

Nom\_champ1 : type

Nom\_champ2 = enregistrement

Champ1: type

Champ2: type

finenregistrement

Nom\_champ3 : type

Finenregistrement

Type:

Nom\_champ2 = enregistrement

Champ1: type

Champ2: type

Finenregistrement

Nom\_type = enregistrement

Nom\_champ1 : type

Nom\_champ : Nom\_champ2

Finenregistrement

Instanciation :

Var :

Objet : nom\_type

Accès aux champ d'un enregistrement (2 Méthodes) :

-Objet.nom\_champ(get/set)

-avec objet faire

Nom\_champ(get/set)

Finavec

## **Chap.8 : Les fichiers**

### **Manipulation des fichiers:**

Associer(nom de variable,"chemin du fichier")

Créer(nom de variable)

Ouvrir(nom de variable,mode:

Lecture : On ne pourra faire que la lecture

Ecriture : Le contenu antérieur sera écrasé (On utilise la commande Ecrire)

Ajout : Le contenu antérieur est préservé (On utilise la commande Ajouter)

)

Fermer(nom de variable)

Supprimer(nom de variable)

### *Fonctions importantes*

*Existe(nom variable) => vrai si le fichier existe*

*EOF(nom variable) => vrai si le prochain char est EOF*

*EOLN(nom variable) => vrai si le prochain char est EOL*

*Ecrire(nom variable, chaine/enregistrement)*

*Ajouter(nom variable, chaine/enregistrement)*

*Lire(nom variable, char/enregistrement)*

*Lireln(nom,variable, chaine)*

Parcours séquentiel:

Tant que non EOF(nom variable)

### **Fichiers à accès séquentiel :**

Fichier texte :

Déclaration :

Nom variable : fichier séquentiel

Fichier binaire :

Déclaration:

Nom variable : fichier séquentiel de  
type(t\_personne,entier...)

Il est impossible de:

modifier directement un enregistrement,  
ajouter un nouvel enregistrement ailleurs qu'à la fin,  
supprimer physiquement un enregistrement.

Ces opérations de mise à jour ne peuvent se faire qu'en réécrivant complètement le fichier.

Pas de modification directe : Pour modifier ou insérer un enregistrement, il faudra tout réécrire

La mise à jour directe n'est possible qu'avec les fichiers à accès direct

### **Fichier à accès direct : (Optimisé plus pour les fichiers binaires)**

Déclaration :

Nom variable : Fichier direct de type(t\_personne,...)

Manipulation :

Index de 1 => taille(nom variable)

Associer(nom variable, "chemin du fichier")

Créer(nom variable)

Ouvrir(nom variable) // *Pas de mode d'ouverture*

Lire(nom variable, **index, variable ou on va stocker l'enregistrement**)

Ecrire(nom variable,**index, enregistrement**)

Ajouter(nom variable,enregistrement)

*Si on connaît l'indice de l'enregistrement qu'on veut traiter, on pourra l'accéder directement.*

*Sionon, on fait un parcours séquentiel (EOF..)*

### **Fichiers à accès indexé : Uniquement Binaires**

Relire TD fichiers