



Méthodes ASD : Les Tableaux, matrices et enregistrements.

1.les algorithmes de tri :

```
PROCEDURE TriSélection(VAR T :TABL; TAILLE :ENTIER)
VAR i,pos :ENTIER
DÉBUT
POUR i DE 1 A TAILLE -1
FAIRE
pos ← trouverMin(T,i) // Plus petit élément du tableau
restant
echanger(T,i,pos) // Echanger avec l'élément en position
i
FINPOUR
FIN

// Fonction retournant l'indice du plus petit élément dans le tableau entre les indices i
et la fin du tableau

FONCTION trouverMin(T:TABL i: ENTIER ):ENTIER
VAR i_Min, j :ENTIER
DÉBUT
i_Min ← i
POUR j DE i_Min +1 A TAILLE
FAIRE
SI T [j] < T[i_Min]
ALORS
i_Min ← j
FINSI
FINPOUR
RETOURNER( i_Min)
FIN
```



// PROCÉDURE échangeant les éléments d'indices i et j dans un tableau

PROCEDURE *Echanger* (VAR T: TABL; i,j: ENTIER)

VAR Aux :ENTIER;

DEBUT

Aux \leftarrow T[i]

T[i] \leftarrow T[j]

T[j] \leftarrow Aux

FIN

2.les algorithmes de recherche:

Recherche Séquentielle

Var i , N : entier

x : valeur

T : Tableau[1..Max] de valeur

Début

i := 1

tant que $i \leq N$ et $T[i] \neq x$ faire

i := i+1

fait

si i > N



```
alors écrire(x,  $\ell$  n'est pas dans T  $\ell$  )

sinon écrire(x,  $\ell$  se trouve dans T à la position  $\ell$  , i)

fsi

Fin

Var g , d, m, N : entier

x : valeur

arrêt : booléen

T : Tableau[1..Max] de valeur
```

Recherche Dichotomique

```
Début

g := 1

d := N

arrêt := faux

Répéter

    m := (g+d) Div 2

    Si x = T[m]

        alors arrêt := vrai

    sinon Si x < T[m]

        alors d := m-1

    sinon g := m+1
```



```
    fsi  
  
    fsi  
  
jqa arrêt ou  $g > d$   
  
si  $g > d$   
  
alors écrire( $x$ ,  $\ell$  n'est pas dans  $T$   $\ell$  )  
  
sinon écrire( $x$ ,  $\ell$  se trouve dans  $T$  à la position  $\ell$  , $m$ )  
  
fsi  
  
Fin
```

3.Insertion d'un élément:

Insertion_tab_non_trié

```
Var N : entier  
  
x : valeur  
  
T : Tableau[1..Max] de valeur  
  
Début  
  
N := N+1  
  
T[N] := x  
  
Fin
```

Insertion_tab_trié

```
Var i , N : entier
```



```
x : valeur

T : Tableau[1..Max] de valeur

Début

i := N

tant que i > 0 et T[i] > x faire

    T[i+1] := T[i]

    i := i-1

fait

T[i+1] := x

N := N+1

Fin
```

4. Suppression d'un élément:

```
Suppression_tab_non_trié (var T : Tab_valeur ; var N : entier  
; x : valeur)
```

```
Var pos : entier
```

```
Début
```

```
pos := Recherche_Séquentielle(T,N,x)
```

```
si pos ≠ 0
```

```
alors T[pos] := T[N]
```



```
N := N-1
```

```
fsi
```

```
Fin
```

```
Suppression_tab_trié (var T : Tab_valeur ; var N : entier ; x  
: valeur)
```

```
Var i : entier
```

```
Début
```

```
i := 1
```

```
tant que i ≤ N et T[i] ≠ x faire
```

```
i := i+1
```

```
fait
```

```
si i ≤ N
```

```
alors T[i] := T[N]
```

```
N := N-1
```

```
fsi
```

```
Fin
```

Remarque :

On peut remarquer que le même travail peut se faire pour une matrice en remplaçant $t[i]$ par $t[i,j]$ et utiliser une autre boucle (compteur j) pour parcourir les lignes.



Exemple

Fonction recherche (M:Mat ; n,x:Entier):boolean

 i,j:entier

 ok:boolean

Début

 i<--1

 ok<--faux

 Tant que (i<=n) et non(ok) faire

 j<--1

 Tantque (j<=n) et non(ok) faire

 si (M[i,j]=x) alors

 ok<--vrai

 sinon

 j<--j+1

 FinSi

 Fin Tantque

 i<--i+1

Fin Tantque

recherche<--ok



Fin

Les types enregistrements font partie des types structurés.

Concernant les champs,

Il faut savoir que ceux-ci peuvent être de n'importe quel type (sauf un type fichier).

Mais :

- Il n'existe pas de constante d'un type enregistrement.
- Il n'existe pas d'opération (autre que l'affectation et le passage comme paramètre) sur les enregistrements.
- Les seules expressions d'un type enregistrement sont les variables de ce type.
- Il n'y a pas de fonction (même prédéfinie) à résultat d'un type enregistrement.

Accès au champ d'un enregistrement

Alors que les éléments d'un tableau sont accessibles au travers de leur indice, les champs d'un enregistrement sont accessibles à travers leur nom, grâce à l'opérateur '.'

Un tel champ est défini par le nom de l'enregistrement ainsi que par son nom propre.

`apprenant.nomprenom`: représente le champ `nomprenom` de l'enregistrement `apprenant`.

Le nom d'un champ est **TOUJOURS** précédé du nom de l'enregistrement auquel il appartient.

On ne peut pas trouver un nom de champ tout seul, sans indication de l'enregistrement.

Les champs d'un enregistrement, tout comme les éléments d'un tableau, sont des variables à qui on peut faire subir les mêmes opérations (affectation, saisie, affichage,...).

Affectation

L'affectation de valeurs aux différents champs d'une variable enregistrement se fait comme suit :

```
Variable.champ<--- valeur
```

Lecture

La lecture des valeurs des différents champs d'une variable enregistrement se fait comme suit :



Lire (Variable.champ)

Écriture

L'écriture des valeurs des différents champs d'une variable enregistrement se fait comme suit :

Ecrire (Variable.champ)



easy ways