ALGO ET PROG FICHE - 1

ALGORITHMES USUELS SUR LES TABLEAUX

1 Initialisation par saisie

On veut faire remplir n cases d'un tableau de MAX cases par l'utilisateur.
Les cases devront contenir des valeurs réelles entre -100.0 et 100.0 par exemple.

```
/* avec filtrage des saisies */
                                        Rappel :
CONST MAX = 300 ;                       t Tableau
TYPE Tab = Tableau[MAX] de Réel ;       i Indice
                                        t[i] Valeur dans la case i du tableau t
Programme Saisie_A
    VAR n, i : Entier ;
        t    : Tab ;
début
    Ecrire("Combien de valeurs ? 0 à ", MAX, " : ");
    Lire(n) ;

    pour i de 0 à (n-1) pas 1 faire
     | Ecrire("Valeur ? ") ;
     | Lire(t[i]);
    fin pour
fin
```

```
/* avec filtrage des saisies */

CONST MAX = 300 ;
TYPE Tab = Tableau[MAX] de Réel ;

Programme Saisie_B
    VAR n, i : Entier ;
        t    : Tab ;
début
    Ecrire("Combien de valeurs ? 0 à ", MAX, " : ");
    Répéter
     | Lire(n) ;
    jusqu'à (n ≥ 0 ET n ≤ MAX);

    pour i de 0 à (n-1) pas 1 faire
     | Ecrire("Valeur ? ") ;
     | Répéter
     |  | Lire(t[i]);
     | jusqu'à (t[i] ≥ -100.0 ET t[i] ≤ 100.0);
    fin pour
fin
```

2 Valeur maximale

On veut obtenir la valeur maximale se trouvant dans le tableau.
PRECONDITION : le tableau contient au moins une case (n > 0) au vu du prog. de saisie.

```
Programme Maxi_tableau_NON_VIDE
    VAR n, i : Entier ;
        t    : Tab ;
        maxi : Réel ;       /* Rajout à faire dans l'en-tête */
début
    ...                              /* même code que le programme de saisie  */

    maxi ← t[0] ;                   /* VALEUR en 0 prise comme max temporaire */

    pour i de 1 à (n-1) faire         /* puis on cherche dans les suivantes */
     | si (t[i] > maxi) alors
     |  | maxi ← t[i] ;
     | fin si
    fin pour

    Ecrire("Valeur maximale : ", maxi) ;
fin
```

3 Position de la valeur maximale

On veut l'indice de la première valeur correspondant à la valeur maximale.
PRECONDITION : le tableau contient au moins une case (n > 0) au vu du prog. de saisie.

```
Programme indice_Maxi_tableau_NON_VIDE
    VAR n, i, imaxi : Entier ;      /* Rajout à faire dans l'en-tête */
        t    : Tab ;
début
    ...                              /* même code que le programme de saisie  */

    imaxi ← 0 ;                    /* INDICE 0 pris comme position temporaire */

    pour i de 1 à (n-1) faire         /* puis on cherche dans les suivantes */
     | si (t[i] > t[imaxi]) alors
     |  | imaxi ← i ;
     | fin si
    fin pour

    Ecrire("Premier indice contenant le maximum : ", imaxi) ;
fin
```

On obtient le premier indice contenant la valeur maximale si plusieurs cases contiennent cette valeur car on change l'indice `imaxi` uniquement lorsqu'on trouve strictement plus que la valeur maximale `t[imaxi]`.

Variante : pour trouver la dernière case contenant le maximum, on utilise "supérieur ou égal" :

```
     | si (t[i] ≥ t[imaxi]) alors
```

4 Valeur minimale

On veut obtenir la valeur minimale se trouvant dans le tableau.
PRECONDITION : le tableau contient au moins une case (n > 0) au vu du prog. de saisie.

```
Programme Mini_tableau_NON_VIDE
    VAR n, i : Entier ;          /* Rajout à faire dans l'en-tête */
        t           : Tab ;
        mini : Réel ;       /* Rajout à faire dans l'en-tête */
début
    ...                           /* même code que le programme de saisie  */

    mini ← t[0] ;                /* VALEUR en 0 prise comme min temporaire */

    pour i de 1 à (n-1) faire         /* puis on cherche dans les suivantes */
     | si (t[i] < mini) alors
     |   | mini ← t[i] ;
     | fin si
    fin pour

    Ecrire("Valeur minimale : ", mini) ;
fin
```

5 Position de la valeur minimale

On veut l'indice de la première valeur correspondant à la valeur minimale.
PRECONDITION : le tableau contient au moins une case (n > 0) au vu du prog. de saisie.

```
Programme indice_Mini_tableau_NON_VIDE
    VAR n, i, imini : Entier ;      /* Rajout à faire dans l'en-tête */
        t    : Tab ;
début
    ...                           /* même code que le programme de saisie  */

    imini ← 0 ;                /* INDICE 0 pris comme position temporaire */

    pour i de 1 à (n-1) faire         /* puis on cherche dans les suivantes */
     | si (t[i] < t[imini]) alors
     |   | imini ← i ;
     | fin si
    fin pour

    Ecrire("Premier indice contenant le minimum : ", imini) ;
fin
```

On obtient le premier indice contenant la valeur minimale si plusieurs cases contiennent cette valeur car on change l'indice `imini` uniquement lorsqu'on trouve strictement moins que la valeur minimale `t[imini]`.

Variante : pour trouver la dernière case contenant le minimum, on utilise "inférieur ou égal" :

```
     | si (t[i] ≤ t[imini]) alors
```

6 Somme et moyenne (pour toutes les valeurs du tableau)

On veut obtenir la moyenne des valeurs se trouvant dans le tableau.
PRECONDITION : le tableau contient au moins une case (n > 0) au vu du prog. de saisie.

```
Programme Moyenne
    VAR n, i : Entier ;
        t    : Tab ;
        somme, moyenne : Réel ; /* Rajout */
début
    ...                              /* même code que le programme de saisie  */

    somme ← 0.0 ;                                /* initialisation importante  */

    pour i de 0 à (n-1) faire                        /* pour toutes les cases */
     | somme ← somme + t[i] ;
    fin pour

    moyenne ← somme / n ;              /* précondition n > 0 donc ok */
    Ecrire("Moyenne : ", moyenne) ;
fin
```

7 Dénombrement

On veut le nombre de valeurs supérieures à zéro.

```
Programme Denombrement_valeurs_positives
    VAR n, i,  nbr : Entier ;    /* rajout */
        t    : Tab ;
début
    ...                              /* même code que le programme de saisie  */

    nbr ← 0 ;                                  /* initialisation importante  */

    pour i de 0 à (n-1) faire                        /* pour toutes les cases */
     | si (t[i] ≥ 0) alors
     |  | nbr ← nbr + 1 ;
     | fin si
    fin pour

    Ecrire("Nombre de valeurs positives : ", nbr) ;
fin
```

8 Moyenne des valeurs positives uniquement

Un exemple un peu plus complexe combinant deux algorithmes basiques :

- Dénombrement

- Moyenne

> On veut la moyenne des valeurs positives.
> On doit donc compter le nombre de valeurs positives et faire leur somme.
> Attention à la division : il est possible d'avoir 0 valeur positive.

```
Programme Moyenne
    VAR n, i, nbr : Entier ;       /* Rajout */
        t     : Tab ;
        somme, moyenne : Réel ; /* Rajout */
début
    ...                            /* même code que le programme de saisie  */

    somme ← 0.0 ;                          /* initialisation importante  */
    nbr ← 0 ;

    pour i de 0 à (n-1) faire                    /* pour toutes les cases */
     | si (t[i] ≥ 0) alors
     |  | nbr ← nbr + 1 ;
     |  | somme ← somme + t[i] ;
     | fin si
    fin pour

    si (nbr > 0) alors
     | moyenne ← somme / nbr ;
    sinon
     | moyenne ← 0 ;
     | Ecrire("Warning : aucune valeur positive") ;
    fin si

    Ecrire("Moyenne : ", moyenne) ;
fin
```

9 Détection de présence

On veut savoir si un ensemble de valeurs contient au moins une valeur négative.
On peut donc stopper et répondre dès qu'on trouve une valeur négative.

```
Programme contient_un_negatif
    VAR n, i       : Entier ;
        t          : Tab ;
        detection : Booléen ;            /* Rajout */
début
    ...                              /* même code que le programme de saisie  */

    detection ← Faux ;                          /* initialisation importante  */
    i ← 0 ;

    /* TQ qu'on a pas détecté de négatif et qu'il reste des cases */
    tant que (NON detection ET i < n) faire
     | si (t[i] < 0) alors
     |  | detection ← Vrai ;
     | fin si
     | i ← i + 1 ;
    fin tant que

    si (detection) alors
     | Ecrire("Il y a au moins une valeur négative") ;
    sinon
     | Ecrire("Aucune valeur négative") ;
    fin si

fin
```