

ALGORITHMES USUELS SUR LES BOUCLES

1 Connaissances fondamentales

VARIABLES

```
/* Affectation à 50 */  
x ← 50 ;  
  
/* Incrémentation / rajoute de 50 */  
x ← x + 50 ;  
  
/* Lire deux valeurs au clavier */  
Lire(x, y) ;  
  
/* Afficher les valeurs à l'écran */  
Ecrire(x, y)
```

ALTERNATIVE

```
si (condition) alors  
    ...  
sinon  
    ...  
fin si
```

COMPLEMENTARITE

```
NON (x ≥ 0) --> (x < 0)  
  
NON arret      --> poursuite  
  
NON poursuite  --> arret
```

BOUCLES

```
tant que (condition poursuite) faire  
    ...  
fin tant que  
  
faire  
    ...  
tant que (condition poursuite) ;  
  
répéter  
    ...  
jusqu'à (condition arrêt);  
  
pour i de 0 à (n-1) pas 1 faire  
    ...  
fin pour  
  
pour i de 1 à n pas 2 faire  
    ...  
fin pour  
  
pour i de n à 1 pas -2 faire  
    ...  
fin pour
```

QUOTIENT et RESTE

```
a DIV b      --> quotient de a / b  
a MOD b      --> reste de a / b  
(a MOD b) = 0 --> a divisible par b
```

A COMPRENDRE : (condition) est à comprendre comme (condition est évaluée Vrai). Il faut parvenir à faire cette transformation dans votre tête :

```
tant que (condition poursuite) faire  
tant que (la condition de poursuite est vraie) faire le bloc d'actions
```

2 Valeur maximale parmi n valeurs

- METHODE 1 : on demande n puis on stocke une valeur dans chacune des n cases.
- METHODE 2 : on récupère les caractères un par un jusqu'à obtenir une valeur servant d'identificateur de fin.
- METHODE 3 : on récupère un caractère puis on demande si on doit continuer.

2.1 METHODE 1 - Valeur maximale parmi n valeurs

On doit

- demander le nombre n de valeurs
- faire la saisie ces n entiers
- afficher la valeur maximale.

```
Programme maximum
  VAR n, i : Entier ;
      maxi, valeur : Réel ;
début
  Ecrire("Combien de valeurs voulez-vous rentrer ?") ;
  répéter
    Lire(n) ;
  jusqu'à (n > 0);

  Ecrire("Donner une valeur : ");
  Lire(valeur);
  maxi ← valeur ; /* valeur initiale prise comme max temporaire */

  pour i de 1 à (n-1) faire /* demande les (n-1) suivantes */
  | Ecrire("Donner une valeur : ");
  | Lire(valeur) ;
  | si (valeur > maxi) alors
  | | maxi ← valeur ;
  | fin si
  fin pour

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

2.2 METHODE 2 - Valeur maximale avec question o/n

On veut saisir des entiers **quelconques** SANS demander à l'utilisateur combien de valeurs il veut donner.

On doit donc demander à l'utilisateur s'il veut continuer ou pas à chaque fois qu'on récupère une de ses valeurs.

```
Programme maximum2
  VAR n, i : Entier ;
      maxi, valeur : Réel ;
      continuer : Caractère ;
début
  Ecrire("Donner une valeur :");
  Lire(valeur);
  maxi ← valeur ;      /* valeur initiale prise comme max temporaire */

  Ecrire("Voulez-vous continuer ? (o/n) ? ");
  Lire(continuer) ;

  tant que (continuer = 'o')          /* puis on demande les suivantes */
  |   Ecrire("Donner une valeur : ");
  |   Lire(valeur) ;
  |   si (valeur > maxi) alors
  |   |   maxi ← valeur ;
  |   fin si
  |   Ecrire("Voulez-vous continuer ? (o/n) ? ");
  |   Lire(continuer) ;
  fin tant que

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

2.3 METHODE 3 - Valeur maximale avec indicateur de fin

On veut saisir des entiers **positifs** SANS DEMANDER à l'utilisateur ni le nombre de valeurs, ni s'il veut continuer.

On encode le fait de vouloir arrêter par le fait de saisir la valeur **-1**.

Ce -1 n'est donc pas une valeur possible à transmettre, il joue un rôle de délimitateur. Par contre, -1 ne pouvant plus être

```
Programme maximum3
  VAR n, i : Entier ;
      maxi, valeur : Réel ;
début
  Ecrire("Donner une valeur positive.");
  Répéter
    Lire(valeur);          /* on veut une valeur positive ! */
  jusqu'à (valeur ≥ 0);
  maxi ← valeur ;          /* valeur initiale prise comme max temporaire */

  tant que (valeur ≠ -1)    /* puis on demande les suivantes */
  | Ecrire("Donner une valeur positive (-1 pour finir) : ");
  | répéter
  | | Lire(valeur) ;
  | | jusqu'à (valeur ≥ -1);          /* on ne veut pas -2 ou moins */
  | | si (valeur ≠ -1 ET valeur > maxi) alors
  | | | maxi ← valeur ;
  | | fin si
  fin tant que

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

3 Position de la valeur maximale

On peut encore le faire des trois manières. On ne présentera ici que la METHODE 1.
On demande combien d'entiers on veut, on rentre les valeurs et le système donne le numéro de la valeur qui correspond à ce maximum.

Exemple : vous rentrez 10, **15**, 12.

Le système écrit 1 car c'est l'indice du 15.

En numérotation "humaine", nous aurions dit 2.

```
Programme position_Maxi
  VAR n, numero_du_maxi : Entier ;
      valeur, maxi : Réel ;
début
  Ecrire("Combien de valeurs voulez-vous rentrer ?") ;
  répéter
    Lire(n) ;
  jusqu'à (n > 0);

  Ecrire("Donner une valeur : ");
  Lire(valeur);
  maxi ← valeur ; /* valeur initiale prise comme max temporaire */
  numero_du_maxi ← 0 /* numérotation comme dans un tableau */

  pour i de 1 à (n-1) faire /* demande les (n-1) suivantes */
    Ecrire("Donner une valeur : ");
    Lire(valeur) ;
    si (valeur > maxi) alors
      maxi ← valeur ;
      indice_du_maxi ← i ;
    fin si
  fin pour

  Ecrire("Indice de la valeur maximale : ", indice_du_maxi) ;
  Ecrire("Position humaine de la valeur ", indice_du_maxi + 1) ;
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 (valeur >= maxi) alors
```

4 Position de la valeur minimale

Comme ci-dessus mais avec le plus petit. Exemple : vous rentrez **10**, 15, 12.
Le système écrit 0 car c'est l'indice du 10. En numérotation "humaine", nous aurions dit 1.

```
Programme position_Mini
  VAR n, numero_du_mini : Entier ;
      valeur, mini : Réel ;
début
  Ecrire("Combien de valeurs voulez-vous rentrer ?") ;
  répéter
    | Lire(n) ;
  jusqu'à (n > 0);

  Ecrire("Donner une valeur : ");
  Lire(valeur);
  mini ← valeur ; /* valeur initiale prise comme min temporaire */
  numero_du_mini ← 0 /* numérotation comme dans un tableau */

  pour i de 1 à (n-1) faire /* demande les (n-1) suivantes */
    | Ecrire("Donner une valeur : ");
    | Lire(valeur) ;
    | si (valeur < mini) alors
      | | mini ← valeur ;
      | | indice_du_mini ← i ;
    | fin si
  fin pour

  Ecrire("Indice de la valeur minimale : ", indice_du_mini) ;
  Ecrire("Position humaine de la valeur ", indice_du_mini + 1) ;
fin
```

5 Somme et moyenne (de toutes les valeurs rentrées)

On veut obtenir la moyenne des valeurs **positives** rentrées. On utilise la METHODE 3 (-1 pour signaler la fin).

```
Programme moyenne
  VAR n, i : Entier ;
      valeur, \enrouge{somme}, \enrouge{moyenne} : Réel ;
début
  somme ← 0 ;
  n ← 0;

  Ecrire("Donner une valeur positive.");
  Répéter
  | Lire(valeur); /* on veut une valeur positive ! */
  | si (valeur ≥ 0) alors
  | | somme ← somme + 1 ;
  | | n ← n + 1 ;
  | fin si
  jusqu'à (valeur = -1);

  si (n > 0) alors
  | moyenne ← somme / n;
  | Ecrire("Moyenne des valeurs tapées : ", moyenne);
  sinon
  | Ecrire("Aucune moyenne car aucune valeur !");
  fin si
fin
```

6 Dénombrement

On veut le nombre de valeurs saisies supérieures à zéro.
On utilisera LA METHODE 2 (question o/n) pour la saisie.

```
Programme denombrement
  VAR nbr : Entier ;
      valeur : Réel ;
      continuer : Caractère ;
début
  nbr ← 0;

  continuer ← 'o' ;
  tant que (continuer = 'o')          /* tant que oui */
  |   Ecrire("Donner une valeur : ");
  |   Lire(valeur) ;
  |   si (valeur > 0) alors
  |   |   nbr ← nbr + 1 ;
  |   fin si
  |   Ecrire("Voulez-vous continuer ? (o/n) ? ");
  |   Lire(continuer) ;
  fin tant que

  Ecrire("Nombre de valeurs > 0 : ", nbr) ;
fin
```