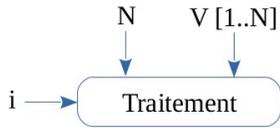
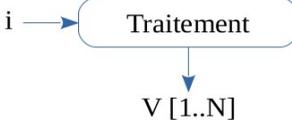
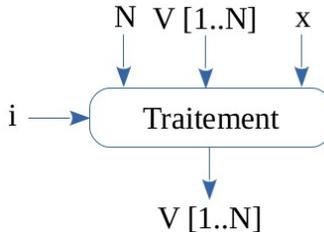
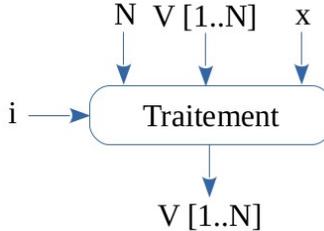


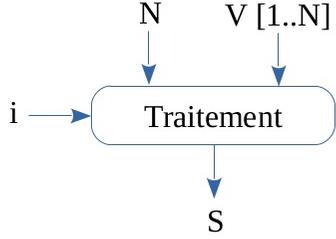
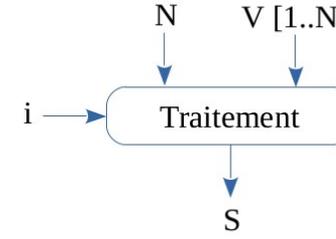
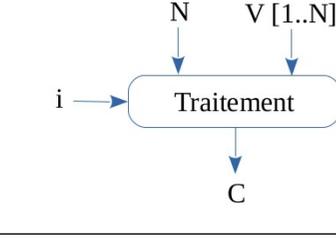
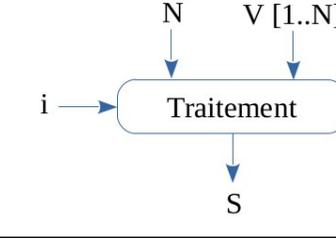
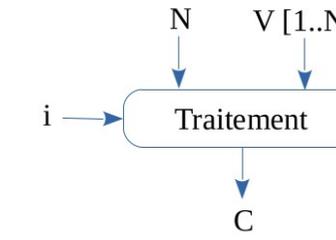
Traitements sur les Vecteurs & Matrices (Tableaux – Variables indicées)

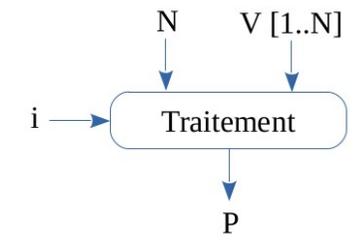
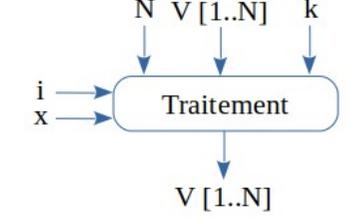
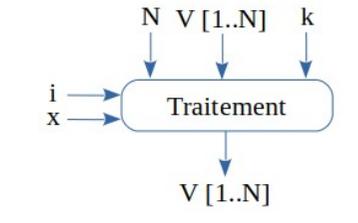
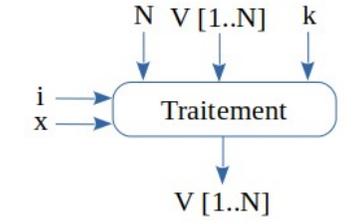
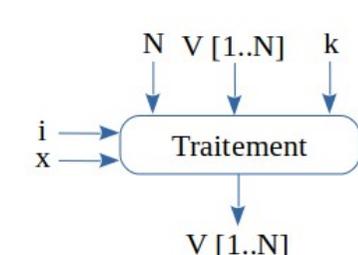
Partie - 01 - Les Vecteurs

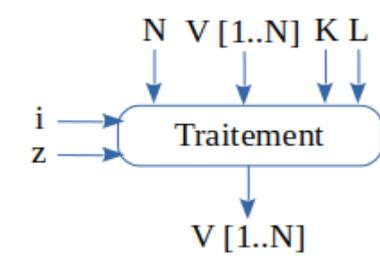
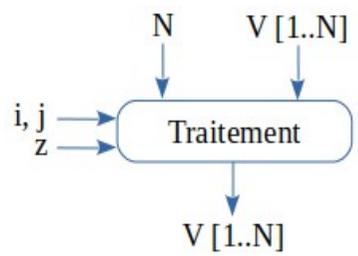
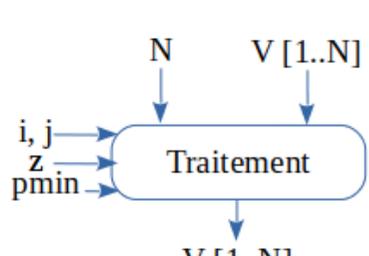
Comment réaliser différentes opérations sur les vecteurs en algorithmique

-*- Tableaux à une Dimension: Vecteurs -*-

| Opérations | Schéma (Analyse) | Algorithmique (Traitement) |
|---|--|---|
| Déclaration d'un vecteur V de type réel et d'ordre N | / | V : Tableau [1..100] de Réel; N, i : entier; |
| Lecture d'un vecteur V - Introduire sa taille N - Introduire les cases V[i] Link : https://onlinegdb.com/zF26AW5UN |  | Lire (N); Pour i ← 1 à N faire Lire (V[i]); Fin-Pour ; |
| Affichage d'un vecteur V - Afficher les cases V[i] / i = 1...N Link : https://onlinegdb.com/bcTyb1NpC |  | Pour i ← 1 à N faire Écrire (V[i]); Fin-Pour ; |
| Ajouter une même valeur x à tous les éléments du vecteur V. Link : https://onlinegdb.com/5vCBf8OUB |  | Pour i ← 1 à N faire V[i] ← V[i] + x; Fin-Pour ; |
| Ôter une même valeur x de tous les éléments du vecteur V. Link : https://onlinegdb.com/KCGYmnOGq |  | Pour i ← 1 à N faire V[i] ← V[i] - x; Fin-Pour |

| | | |
|---|--|--|
| <p>Faire la somme de tous les composants du vecteur.</p> <p>Link : https://onlinegdb.com/ZzID8oScy</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> S </pre> | <pre> S ← 0; Pour i ← 1 à N faire S ← S + V[i]; Fin-Pour; </pre> |
| <p>Faire la somme de tous les composants du vecteur V divisibles par 3.</p> <p>Link : https://onlinegdb.com/76b3U6C_K</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> S </pre> | <pre> S ← 0; Pour i ← 1 à N faire Si V[i] mod 3 = 0 alors S ← S + V[i]; Fin-Si; Fin-Pour; </pre> |
| <p>Compter le nombre des composants du vecteur V divisibles par 3.</p> <p>Link : https://onlinegdb.com/5MmIx713l</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> C </pre> | <pre> c ← 0; Pour i ← 1 à N faire Si V[i] mod 3 = 0 alors c ← c + 1; Fin-Si; Fin-Pour; </pre> |
| <p>Faire la somme de tous les composants du vecteur V divisibles par 3 et non divisibles par 4.</p> <p>Link : https://onlinegdb.com/DD_8n_1VF</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> S </pre> | <pre> S ← 0; Pour i ← 1 à N faire Si (V[i] mod 3 = 0) ET (V[i] mod 4 <> 0) alors S ← S + V[i]; Fin-Si; Fin-Pour; </pre> |
| <p>Compter le nombre des composants du vecteur V divisibles par 3 et non divisibles par 4.</p> <p>Link : https://onlinegdb.com/MWlnULAHS</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> C </pre> | <pre> c ← 0; Pour i ← 1 à N faire Si (V[i] mod 3 = 0) ET (V[i] mod 4 <> 0) alors c ← c + 1; Fin-Si; Fin-Pour; </pre> |

| | | |
|---|--|--|
| <p>Faire le produit de tous les composants du vecteur</p> <p>Link : https://onlinegdb.com/QZBSqvoec</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> P </pre> | <pre> P ← 1; Pour i ← 1 à N faire P ← P * V[i]; Fin-Pour </pre> |
| <p>Ajouter la valeur de la composante d'indice k à tous les éléments du vecteur V</p> <p>Link : https://onlinegdb.com/jEMYgOK8h</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement k --> Traitement i --> Traitement x --> Traitement Traitement --> V[1..N] </pre> | <pre> x ← V[k]; Pour i ← 1 à N faire V[i] ← V[i] + x; Fin-Pour; </pre> |
| <p>Supprimer la valeur de la composante d'indice k à tous les éléments du vecteur V</p> <p>Link : https://onlinegdb.com/sRDpzy61eX</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement k --> Traitement i --> Traitement x --> Traitement Traitement --> V[1..N] </pre> | <pre> x ← V[k]; Pour i ← 1 à N faire V[i] ← V[i] - x; Fin-Pour; </pre> |
| <p>Multiplier la valeur de la composante d'indice k à tous les éléments du vecteur V</p> <p>Link : https://onlinegdb.com/HSXCVAVIt</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement k --> Traitement i --> Traitement x --> Traitement Traitement --> V[1..N] </pre> | <pre> x ← V[k]; Pour i ← 1 à N faire V[i] ← V[i] * x; Fin-Pour; </pre> |
| <p>Diviser la valeur de la composante d'indice k à tous les éléments du vecteur V</p> <p>Link : https://onlinegdb.com/3c758I0tUr</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement k --> Traitement i --> Traitement x --> Traitement Traitement --> V[1..N] </pre> | <pre> x ← V[k]; si x <> 0 then Pour i ← 1 à N faire V[i] ← V[i] / x; Fin-Pour; fin-si; </pre> |

| | | |
|---|---|--|
| <p>Permutation des cases de rang K et L du vecteur V</p> |  | <pre> Z ← V[K]; V[k] ← V[L]; V[L] ← Z; </pre> |
| <p>Tri croissant d'un vecteur V (Ordonner les éléments du vecteur du plus petit vers le plus grand)</p> <p>Méthode 01 : Tris par sélection (permutation(s)) :</p> <ul style="list-style-type: none"> - On recherche toutes les cases avec indice j (j=i+1 à N) tel-que $V[j] < V[i]$ et on permute entre $V[j]$ et $V[i]$ <p><i>Tel-que $i=1 \dots (n-1)$.</i></p> |  | <pre> pour i ← 1 à (n-1) faire pour j ← (i+1) à n faire si V[j] < V[i] alors z ← V[i]; V[i] ← V[j]; V[j] ← z; fin-si; fin-pour; fin-pour; </pre> |
| <p>Tri croissant d'un vecteur V (Ordonner les éléments du vecteur du plus petit vers le plus grand)</p> <p>Méthode 02 : Tris par sélection : Recherche de position & permutation(s) :</p> <ul style="list-style-type: none"> - On cherche l'indice du minimum pmin dans les cases d'indice j / j=(i+1) et N - On permute par la suite en $V[pmin]$ et $V[i]$ <p><i>Tel-que $i=1 \dots (n-1)$.</i></p> |  | <pre> pour i ← 1 à (n-1) faire {Opération 1: chercher pmin : position du min} Pmin ← i; Pour j ← (i+1) à N faire Si V[j] < V[pmin] alors pmin ← j; Fin-Si; Fin-Pour ; {Opération 2:permuter entre les cases pmin et i} Si pmin <> i alors Z ← V[pmin]; V[pmin] ← V[i]; V[i] ← V[Z]; Fin-Si; fin-pour; </pre> |

Link : <https://onlinegdb.com/9n7XyBvOv>

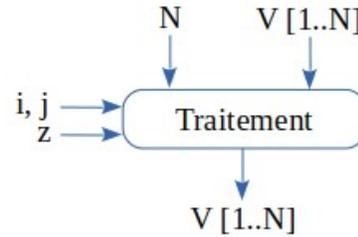
Link : https://onlinegdb.com/88zl_FVGH

Link : <https://onlinegdb.com/CHMp9cZ9V>

Tri croissant d'un vecteur V (Ordonner les éléments du vecteur du plus petit vers le plus grand)

Méthode 03 : Tris par Bulles

Link : https://onlinegdb.com/R_k6WsBhxz



```

Pour j ← 1 à (N-1) faire
  Pour i ← 1 à (N-j) faire
    Si V[i] > V[i+1] alors
      Z ← V[i];
      V[i] ← V[i+1] ;
      V[i+1] ← Z ;
    fin-si;
  Fin-Pour ;
Fin-Pour;
  
```

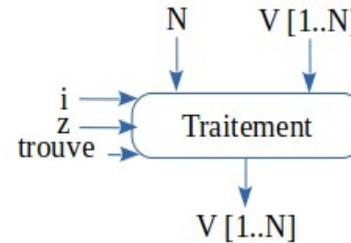
Tri croissant d'un vecteur V (Ordonner les éléments du vecteur du plus petit vers le plus grand)

Méthode 04 : Tris par Bulles (améliorée)

Link : <https://onlinegdb.com/wLZOkkb-2>

Link : <https://onlinegdb.com/6TZG5Ughqh>

Link : https://onlinegdb.com/-HSUmKa_x



```

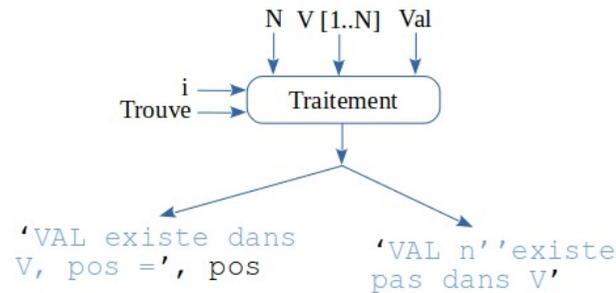
Répéter
  Modifie ← false;
  pour i ← 1 à (N-1) faire
    Si V[i] > V[i+1] alors
      Z ← V[i];
      V[i] ← V[i+1] ;
      V[i+1] ← Z ;
      Modifie ← True;
    Fin-si;
  Fin-Pour;
Jusqu'à (Non (Modifie));
  
```

Rechercher une valeur VAL dans un vecteur V de N éléments et sa position dans V.

Méthode 01 : utiliser une variable booléenne **Trouve**

Link : <https://onlinegdb.com/gobjpCeeX>

Link : <https://onlinegdb.com/6mL7GZPAG>



```

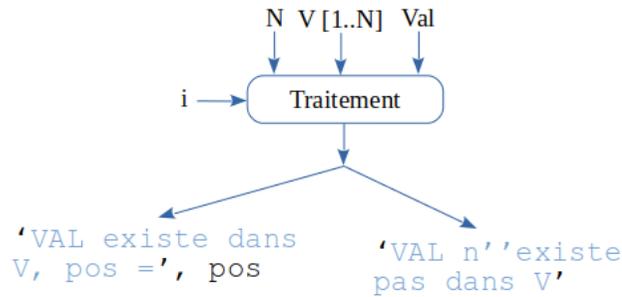
Trouve ← false;
pos ← -1;
i ← 1;
tant-que (i ≤ N) ET (Trouve = false) faire
  si V[i] = VAL alors
    pos ← i;
    trouve ← true;
  fin-si;
  i ← i+1;
fin-tant-que;

si Trouve alors
  écrire('VAL existe dans V, pos', pos);
sinon
  écrire('VAL n''existe pas dans V');
fin-si;
  
```

Rechercher une valeur VAL dans un vecteur V de N éléments et sa position dans V.

Méthode 02 : utiliser uniquement la valeur pos (position)

Link : <https://onlinegdb.com/LCzItdfR5>
 Link : <https://onlinegdb.com/sWB68427eW>



```

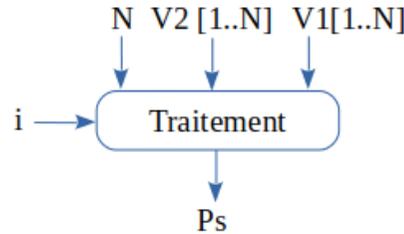
pos ← -1;
i ← 1;
tant-que (i<=N) ET (pos = -1) faire
    si V[i] = VAL alors
        pos ← i;
    fin-si;
    i ← i+1;
fin-tant-que;
    
```

```

si pos <> -1 alors
    écrire('VAL existe dans V, pos=', pos);
sinon
    écrire('VAL n''existe pas dans V');
fin-si;
    
```

Produit Scalaire entre deux vecteurs V1 et V2 de taille N (même taille)

Link :



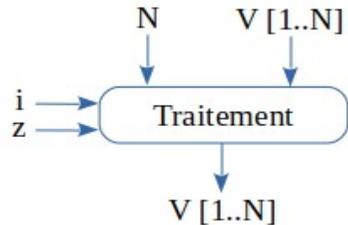
```

Ps ← 0;
pour i ← 1 à n faire
    ps ← ps + V1[i]*V2[i];
fin-pour;
    
```

Rotation à droite des éléments d'un vecteur V : le 1^{er} élément passe au 2^{ème}, le 2^{ème} passe au 3^{ème}, etc.

Si nous avons : V=[1 5 8 9 10]
 nous aurons : V=[10 1 5 8 9]

Link :



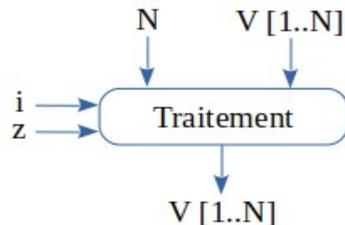
```

Z ← V[n];
pour i ← 1 à (n-1) faire
    V[n-i+1] ← V[n-i];
fin-pour;
V[1] ← Z;
    
```

Rotation à gauche des éléments d'un vecteur V : le 1^{er} élément passe au N^{ème}, le 2^{ème} passe au 1^{ème}, le 3^{ème} passe au 2^{ème}, etc.

Si nous avons : V=[1 5 8 9 10]
 nous aurons : V=[5 8 9 10 1]

Link :

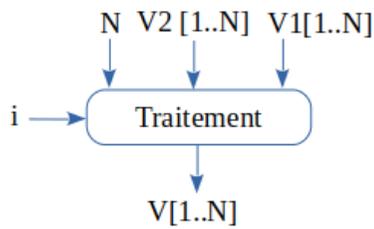


```

Z ← V[1];
pour i ← 1 à (n-1) faire
    V[i] ← V[i+1];
fin-pour;
V[n] ← Z;
    
```

La somme de deux vecteur V1 e V2 de taille N = Vecteur V de taille N
 On écrit : $V = V1 + V2$

Link :

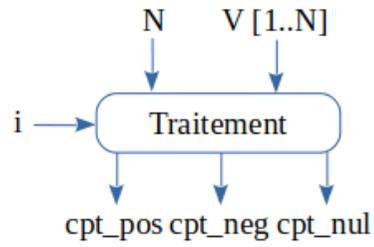


```

pour i ← 1 à n faire
    V[i] ← V1[i] + V2[i];
fin-pour;
    
```

Compter le nombre d'éléments strictement positifs, nombre d'éléments strictement négatif et le nombre d'éléments nuls d'un vecteur V de taille n
 Nous utilisons trois variables entières qui représentent des compteurs :
 cpt_pos , cpt_neg et cpt_nul

Link :

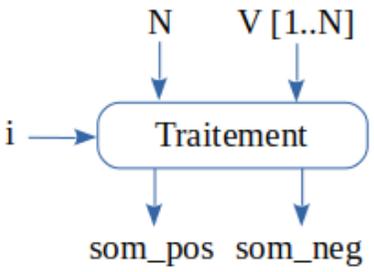


```

cpt_pos ← 0;
cpt_neg ← 0;
cpt_nul ← 0;
pour i ← 1 à n faire
    si V[i] > 0 alors
        cpt_pos ← cpt_pos + 1;
    sinon
        si V[i] < 0 alors
            cpt_neg ← cpt_neg + 1;
        sinon
            cpt_nul ← cpt_nul + 1;
    fin-si;
fin-pour;
    
```

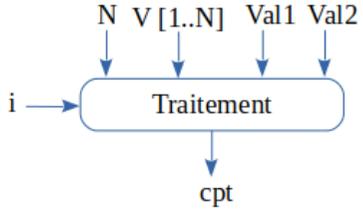
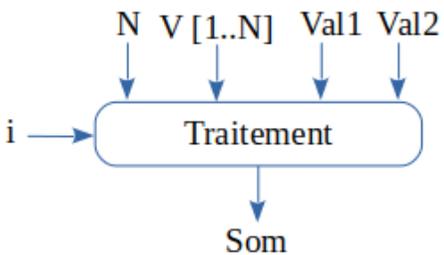
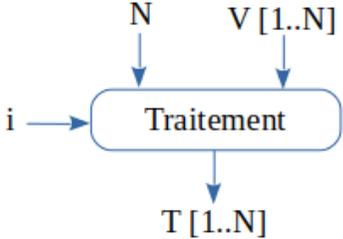
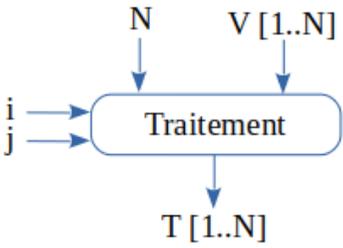
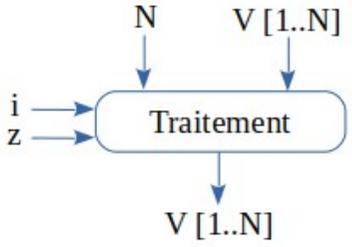
Calculer la somme des éléments positifs et la somme des éléments négatifs d'un vecteur V de taille n.

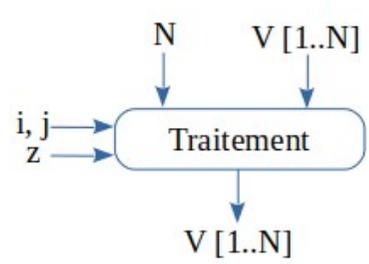
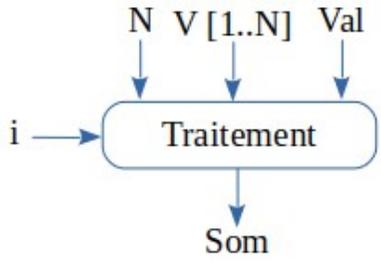
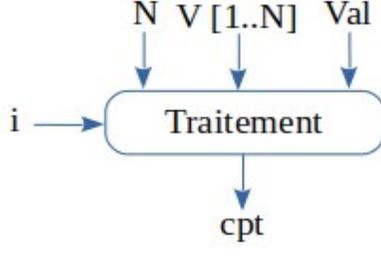
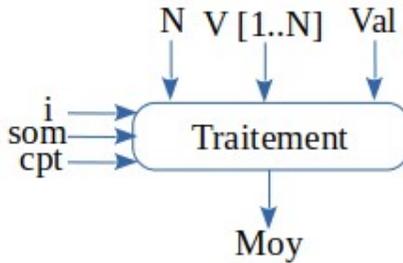
Link :

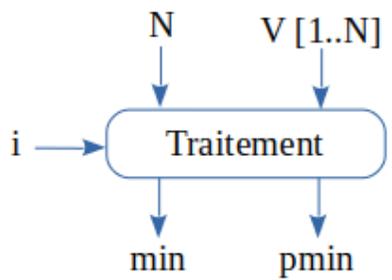
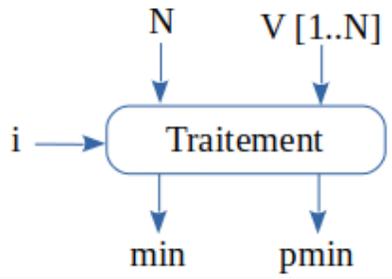
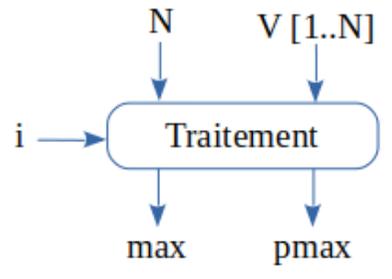
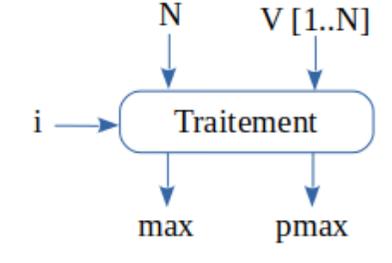
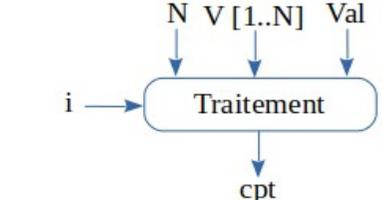


```

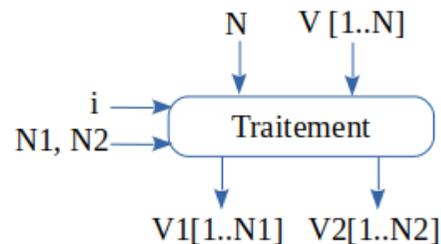
som_pos ← 0;
som_neg ← 0;
pour i ← 1 à n faire
    si V[i] > 0 alors
        som_pos ← som_pos + V[i];
    sinon
        si V[i] < 0 alors
            som_neg ← som_neg + V[i];
        fin-si;
    fin-si;
fin-pour;
    
```

| | | |
|--|--|--|
| <p>Compter le nombre d'éléments d'un vecteur V qui sont entre deux valeurs VAL1 et VAL2, tel-que VAL1 < VAL2.</p> <p>Link :</p> |  | <pre> cpt ← 0; pour i ← 1 à n faire si (V[i] ≥ VAL1) ET (V[i] ≤ VAL2) alors cpt ← cpt + 1; fin-si; fin-pour; </pre> |
| <p>Calculer la somme des éléments d'un vecteur V qui sont entre deux valeurs VAL1 et VAL2, tel-que VAL1 < VAL2.</p> <p>Link :</p> |  | <pre> som ← 0; pour i ← 1 à n faire si (V[i] ≥ VAL1) ET (V[i] ≤ VAL2) alors som ← som + V[i]; fin-si; fin-pour; </pre> |
| <p>Inverser le vecteur V dans un vecteur T de taille N.</p> <p>(Méthode 1)</p> <p>Link :</p> |  | <pre> pour i ← 1 à n faire T[i] ← V[N-i+1]; fin-pour; </pre> |
| <p>Inverser le vecteur V dans un vecteur V de taille N.</p> <p>(Méthode 2)</p> <p>Link :</p> |  | <pre> j ← n; pour i ← 1 à n faire T[i] ← V[j]; j ← j - 1; fin-pour; </pre> |
| <p>Inverser le vecteur V dans lui même</p> <p>(Méthode 1)</p> <p>Link :</p> |  | <pre> pour i ← 1 à (n div 2) faire z ← V[i]; V[i] ← V[N-i+1]; V[N-i+1] ← z; fin-pour; </pre> |

| | | |
|---|--|---|
| <p>Inverser le vecteur V dans lui même (Méthode 2)</p> <p>Link :</p> |  | <pre> j ← N; pour i ← 1 à (n div 2) faire z ← V[i]; V[i] ← V[j]; V[j] ← z; j ← j-1; fin-pour; </pre> |
| <p>La somme des éléments du vecteur V qui sont supérieurs à une valeur VAL.</p> <p>Link :</p> |  | <pre> Som ← 0; pour i ← 1 à n faire si V[i] > VAL alors Som ← Som + V[i]; fin-si; fin-pour; </pre> |
| <p>Le nombre des éléments du vecteur V qui sont supérieurs à une valeur VAL.</p> <p>Link :</p> |  | <pre> Cpt ← 0; pour i ← 1 à n faire si V[i] > VAL alors Cpt ← Cpt + 1; fin-si; fin-pour; </pre> |
| <p>La moyenne des éléments du vecteur V qui sont supérieurs à une valeur VAL.</p> <p>Link :</p> |  | <pre> Cpt ← 0; Som ← 0; pour i ← 1 à n faire si V[i] > VAL alors Cpt ← Cpt + 1; Som ← Som + V[i]; fin-si; fin-pour; si cpt > 0 alors Moy ← Som / Cpt; fin-si; </pre> |

| | | |
|--|--|--|
| <p>La recherche du minimum et sa position dans un vecteur V. (Méthode 1)</p> <p>Link :</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> min Traitement --> pmin </pre> | <pre> min ← V[1]; pmin ← 1; pour i←1 à n faire si V[i]<min alors min ← V[i]; pmin ← i; fin-si; fin-pour; </pre> |
| <p>La recherche du minimum et sa position dans un vecteur V. (Utiliser uniquement l'indice pmin) (Méthode 2)</p> <p>Link :</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> min Traitement --> pmin </pre> | <pre> pmin ← 1; pour i←1 à n faire si V[i]<V[pmin] alors pmin ← i; fin-si; fin-pour; min ← V[pmin]; </pre> |
| <p>La recherche du maximum et sa position dans un vecteur V. (Méthode 1)</p> <p>Link :</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> max Traitement --> pmax </pre> | <pre> max ← V[1]; pmax ← 1; pour i←1 à n faire si V[i] > max alors max ← V[i]; pmax ← i; fin-si; fin-pour; </pre> |
| <p>La recherche du maximum et sa position dans un vecteur V. (Utiliser uniquement l'indice pmax) (Méthode 2)</p> <p>Link :</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement i --> Traitement Traitement --> max Traitement --> pmax </pre> | <pre> pmax ← 1; pour i←1 à n faire si V[i] > V[pmax] alors pmax ← i; fin-si; fin-pour; max ← V[pmax]; </pre> |
| <p>Le nombre d'occurrences d'une valeur VAL dans un vecteur V de taille N Nombre d'occurrences = nombre de répétitions</p> <p>Link :</p> |  <pre> graph TD N --> Traitement V[1..N] --> Traitement Val --> Traitement i --> Traitement Traitement --> cpt </pre> | <pre> cpt ← 0; Pour i←1 à N faire si V[i]=VAL alors cpt ← cpt + 1; Fin-Si; Fin-Pour; </pre> |

Fragmenter un vecteur de type réelle V et de taille N à deux vecteurs $V1$ et $V2$, tel-que $V1$ contient les éléments positifs de V et $V2$ contient les éléments strictement négatifs de V .



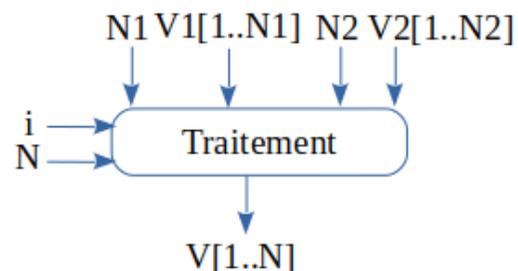
```

N1 ← 0;
N2 ← 0;
Pour i ← 1 à N faire
    Si V[i] < 0 alors
        N1 ← N1 + 1;
        V1[N1] ← V[i];
    Sinon
        N2 ← N2 + 1;
        V2[N2] ← V[i];
    Fin-Si;
Fin-Pour;

```

Link :

Soit deux vecteurs $V1$ et $V2$ de type réel et de taille $N1$ et $N2$ respectivement. Fusionner les deux vecteurs $V1$ et $V2$ dans un autre vecteur V .



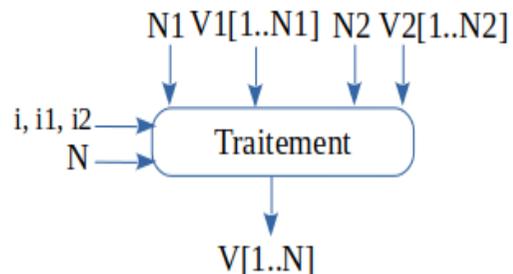
```

Pour i ← 1 à N1 faire
    V[i] ← V1[i];
Fin-Pour;
Pour i ← 1 à N2 faire
    V[N1+i] ← V2[i];
Fin-Pour;
N ← N1+N2;

```

Link : <https://onlinegdb.com/S4mmaDVDj>

Soit deux vecteurs $V1$ et $V2$ de type réel et de taille $N1$ et $N2$ respectivement. Sachant que $V1$ et $V2$ sont triés (ordonnés) avec un ordre croissant, Fusionner les deux vecteurs $V1$ et $V2$ dans un autre vecteur V de tel-sorte que V soit aussi trié avec ordre croissant (sans utiliser les algorithmes de tri).



```

N ← N1+N2;
Pour i ← 1 à N faire
    Si ((i1 <= N1) ET (i2 <= N2)) alors
        Si (V[i1] <= V[i2]) alors
            V[i] ← V1[i1];
            i1 ← i1+1;
        Sinon
            V[i] ← V2[i2];
            i2 ← i2+1;
        Fin-si;
    Sinon
        Si (i1 <= N1) alors
            V[i] ← V1[i1];
            i1 ← i1+1;
        Sinon
            V[i] ← V2[i2];
            i2 ← i2+1;
        Fin-si;
    Fin-Si;
Fin-Pour;

```

Link : <https://onlinegdb.com/7rQKJZQic1>