

Exercice Sup-01: Inverser le vecteur T dans lui même

Ecrire un algorithme/programme PASCAL qui permet d'inverser les éléments d'un vecteur de type réel T dans le même vecteur T (sans utiliser un autre vecteur).

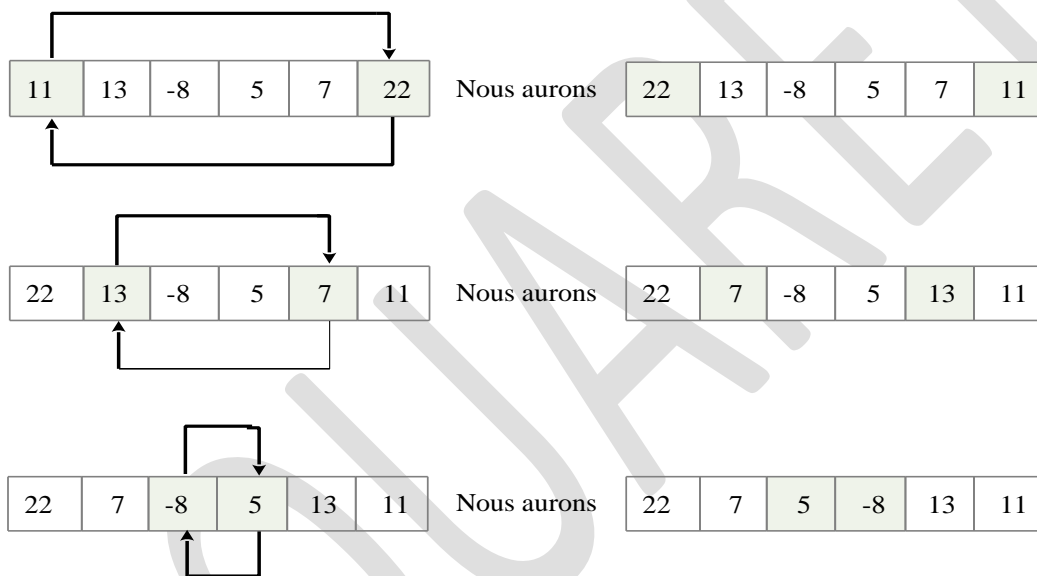
Corrigé de l'exercice Sup-01:

Explication 😊

Pour illustrer la partie traitement, nous prenons un exemple :

$N = 6$, $T = [11 \ 13 \ -8 \ 5 \ 7 \ 22]$

Pour inverser les éléments de T, dans le même vecteur, nous allons faire les permutations suivantes :



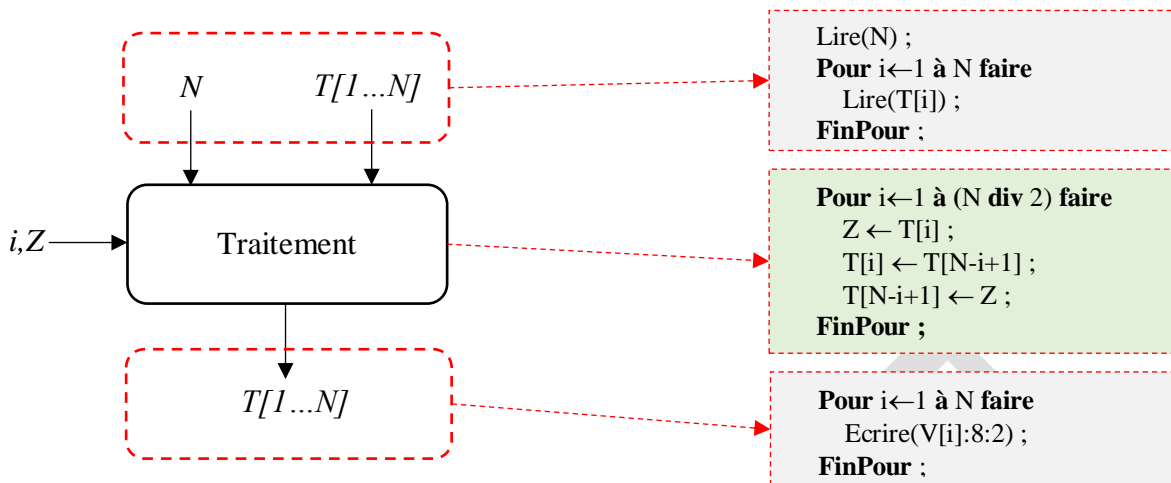
Après la troisième permutation, nous aurons le résultat demandé (inverser le vecteur T dans lui-même).

Remarques à retenir :

- Inverser un vecteur dans lui-même en permutant des cases.
- Le nombre de permutations est la moitié du vecteur.
- Les permutations : (1 avec N), (2 avec N-1), ... Jusqu'à $(N \text{ div } 2)$.
- De la question 1, nous déduisons que : la case $N^\circ i$ sera permutée avec la case $N^\circ (N-i+1)$, tel-que $i = 1 \dots (N \text{ div } 2)$.
- Pour permuter entre les cases i et $(N-i+1)$, nous utilisons une troisième variable Z, comme suit :

```
Z ← T[i]
T[i] ← T[N-i+1]
T[N-i+1] ← Z
Pour chaque valeur de i allant de 1 à (N div 2)
```

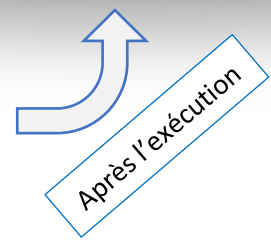
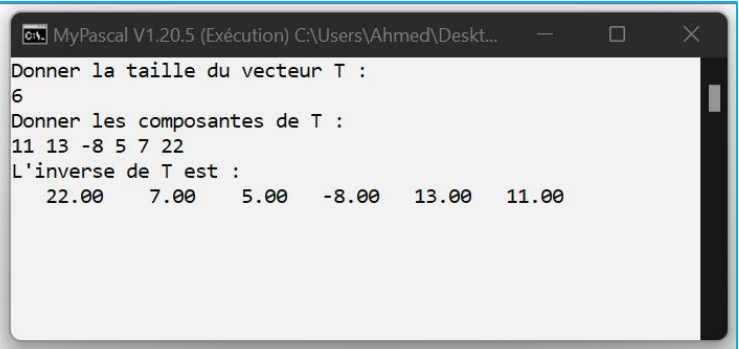
Les variables d'entrée, variable de sortie et la partie traitement sont présentées dans le schéma ci-dessous :



Algorithme/Programme PASCAL

Algorithme	Programme PASCAL
<p>Algorithme inverser_T_dans_T;</p> <p>Variabes <i>i, N</i> : entier; <i>T</i> : Tableau [1..100] de réel; <i>Z</i> : réel;</p> <p>Début</p> <p style="color: red;">{-*-*- Entrées *-*-*-}</p> <p>Ecrire('Donner la taille du vecteur T :');</p> <p>Lire(N);</p> <p>Ecrire('Donner les composantes de T :');</p> <p>Pour <i>i</i>←1 à <i>N</i> faire Lire(T[i]); FinPour ;</p> <p style="color: red;">{-*-*- Traitement *-*-*-}</p> <p>Pour <i>i</i>←1 à (<i>N div 2</i>) faire <i>Z</i> ← T[i]; T[i] ← T[N-i+1]; T[N-i+1] ← <i>Z</i>; FinPour ;</p> <p style="color: red;">{-*-*- Sorties *-*-*-}</p> <p>Ecrire('L'inverse de T est :');</p> <p>Pour <i>i</i>←1 à <i>N</i> faire Ecrire(T[i]:8:2); FinPour ;</p> <p>Fin.</p>	<p>Program inverser_T_dans_T;</p> <p>Var <i>i, N</i> : integer; <i>T</i> : array [1..100] of real; <i>Z</i> : real ;</p> <p>Begin</p> <p style="color: red;">{-*-*- Entrées *-*-*-}</p> <p>Writeln('Donner la taille du vecteur T :');</p> <p>Read(N);</p> <p>Writeln('Donner les composantes de T :');</p> <p>For <i>i</i> := 1 to <i>N</i> do Read(T[i]);</p> <p style="color: red;">{-*-*- Traitement *-*-*-}</p> <p>For <i>i</i> := 1 to (<i>N div 2</i>) do Begin <i>Z</i> := T[i]; T[i] := T[N-i+1]; T[N-i+1] := <i>Z</i>; End;</p> <p style="color: red;">{-*-*- Sorties *-*-*-}</p> <p>Writeln('L'inverse de T est :');</p> <p>For <i>i</i> := 1 to <i>N</i> do Write(T[i]:8:2);</p> <p>End.</p>

```
1 Program inverser_T_dans_T;
2 Var
3   i,N : integer;
4   T : Array [1..100] of real;
5   Z : real;
6 Begin
7   {-*-*- Entrées -*-*-}
8   Writeln('Donner la taille du vecteur T :');
9   Read(N);
10  Writeln('Donner les composantes de T :');
11  For i := 1 to N do
12    Read( T[i] );
13  .....
14  {-*-*- Traitement -*-*-}
15  For i := 1 to (N div 2) do
16    Begin
17      Z := T[i];
18      T[i] := T[N-i+1];
19      T[N-i+1] := Z;
20    End;
21  .....
22  {-*-*- Sorties -*-*-}
23  Writeln('L'inverse de T est :');
24  For i := 1 to N do
25    Write( T[i]:8:2 );
26 End.
```



A. OUAH