

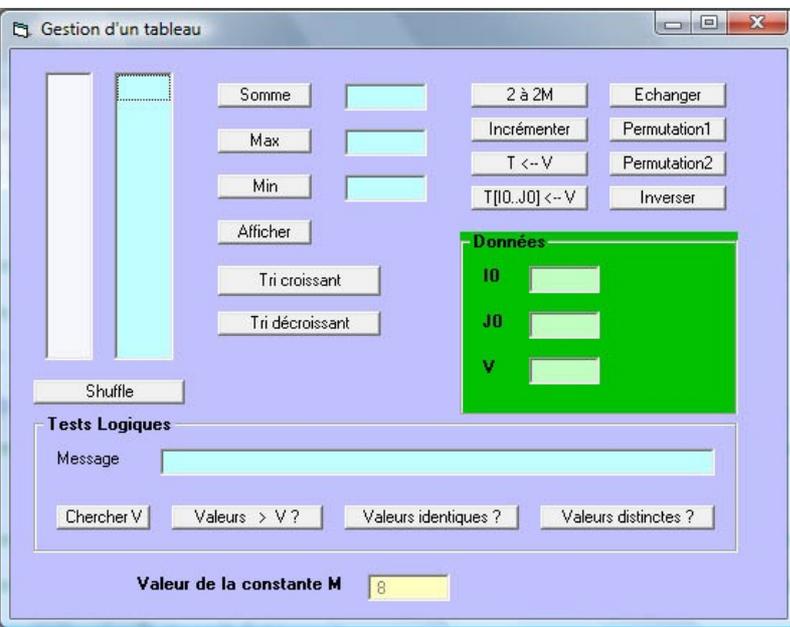
# Gestion d'un tableau d'entier

E.Thirion - 04/07/2015

Les exercices suivants sont en majorité des projets à compléter. L'interface graphique de ces projets est déjà réalisée, ce qui vous permettra un gain de temps important. Ces projets sont disponibles par téléchargement. Pour savoir comment y accéder cliquez [ici](#).

D'autre part, ce document fait partie d'un ensemble de cours du même auteur (programmation procédurale et objet, programmation web, bases de données) auxquels vous pouvez accéder en cliquant [ici](#).

**Projet:** Exo-Tableaux/Pascal/GesTab/ProjetGesTab.lpi

Le formulaire	Zones de texte et zones de liste
	<p style="text-align: center;"><b>Zones de liste</b></p> <p>ZL_Indice ZL_Tableau</p> <p style="text-align: center;"><b>Zones de texte</b></p> <p>ZT_Somme ZT_Max ZT_Min</p> <p>ZT_I0 ZT_J0 ZT_V</p> <p>ZT_Message</p>

## Principe général

Il s'agit de réaliser différentes opérations sur un tableau d'entier.

Les procédures évènementielles associées aux boutons sont déjà écrites. Ne les modifiez pas.

Chacune de ces procédures évènementielle appelle une procédure qui n'est pas encore écrite et que vous devez donc compléter.

## Question 1: Déclaration du tableau

Dans tout l'exercice nous utiliserons le même tableau **N** déclaré en variable globale. Il s'agit d'un tableau d'entiers indicé de 1 à **M** où **M** est une constante.

Il vous faudra donc d'abord déclarer ce tableau. Pour l'instant fixez la valeur de **M** à 8.

## Question 2: Affichage du tableau

La procédure **AfficherTableau** est appelée dans la procédure événementielle du bouton **Afficher**.

Complétez cette procédure afin qu'elle affiche les indices du tableau (de 1 à **M**) dans la zone de liste **ZL\_Indice** et les valeurs correspondantes dans la zone de liste **ZL\_Tableau**.

Pour l'instant, le tableau ne contient que des 0. Vous devriez donc obtenir l'affichage suivant en cliquant sur le bouton **Afficher**:



## Question 3 - Modifications simples du tableau

### Question 3 - A : Le bouton "T <-- V"

Un clic sur ce bouton doit affecter la valeur contenue dans la zone de texte **ZT\_V** à chaque élément du tableau.

Procédure à compléter: **AffecterVaTousLesElementsDuTableau**.

### Question 3 - B : Le bouton "T[I0..J0] <-- V"

Un clic sur ce bouton doit affecter la valeur contenue dans la zone de texte **ZT\_V** aux éléments du tableau dont l'indice est compris entre **I0** (**ZT\_I0**) et **J0** (**ZT\_J0**). Dans l'exemple suivant, l'utilisateur a affecté la valeur 61 aux éléments du tableau dont l'indice est compris entre 3 et 5:



Procédure à compléter: **AffecterV\_DeI0aJ0**

### Question 3 - C : Le bouton Shuffle

Un clic sur ce bouton remplit tableau avec des nombres entiers au hasard compris entre 0 et 9. Rappel: pour un entier  $p$  quelconque, **random** ( $p$ ) retourne un nombre au hasard entre 0 (inclu) et  $p$  (exclu).

Procédure à compléter: **RemplirAuHasard**

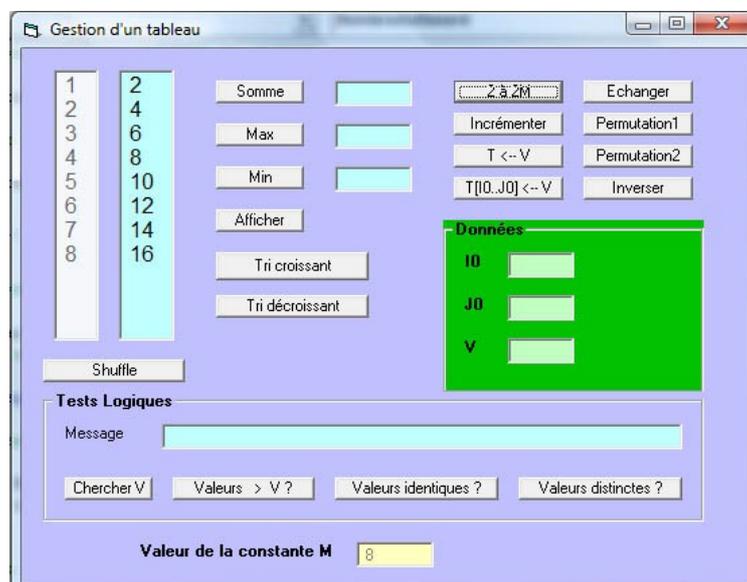
### Question 3 - D : Le bouton "Incrémenter"

Ce bouton ajoute 1 à chaque élément du tableau.

La procédure appelée est **Incrementer**.

### Question 3 - E : Le bouton "2 à 2M"

Un clic sur ce bouton doit remplir le tableau **N** avec les valeurs 2,4,6,..., 2M:



Procédure appelée: **NombresPairs**.

### Question 3 - F : Le bouton "Echanger"

Un clic sur ce bouton doit échanger les valeurs des éléments d'indice **I0** et **J0**.

Procédure appelée: **Echanger** ( $i, j$ ).

Lisez attentivement le code déjà écrit dans la procédure événementielle du bouton.

## Question 4 - Statistiques

### Question 4 - A: Le bouton "Somme"

Lorsque l'utilisateur clique sur ce bouton, le programme affiche la somme des éléments du tableau.

La procédure événementielle associée à ce bouton appelle la fonction **SommeTableau**, que vous devez écrire.

Notez bien que la fonction **SommeTableau** n'affiche rien, elle retourne simplement la valeur de la somme des éléments du tableau **N**.

#### Indications

Utilisez une boucle **for** avec un compteur **i** et une variable **S** qui contiendra à chaque itération la somme des éléments du tableau jusqu'à l'indice **i**.

Avant la boucle, on initialisera la variable **S** à 0.

Exemple (avec  $M = 3$ )

Le tableau

i	1	2	3
N[ i ]	5	-1	4

Déroulement de la boucle for

i	S
	0
1	5
2	4
3	8

Le problème est donc de trouver l'instruction qui doit figurer dans la boucle **for** afin que la variable **S** contienne en permanence la somme des **i** premiers éléments du tableau.

### Question 4 - B : Les boutons "Min" et "Max"

Ces deux boutons permettent respectivement de calculer la plus petite et la plus grande valeur du tableau.

Les procédures événementielles associées à ces boutons appellent respectivement les fonctions **MinTab** et **MaxTab**, que vous devez écrire.

Notez bien que ces deux fonctions n'affichent rien car l'affichage est déjà réalisé dans les procédures événementielles.

#### Indications (pour le Min)

Utilisez une boucle **for** avec un compteur **i** et une variable **Min** qui contiendra à chaque itération le minimum des éléments du tableau jusqu'à l'indice **i**.

Avant la boucle, on initialisera la variable **Min** avec la valeur du premier élément du tableau.

Exemple (avec M=3)Le tableau

i	1	2	3
N[ i ]	5	-1	4

Déroulement de la boucle for

i	Min
	5
1	5
2	-1
3	-1

Le problème est donc de trouver l'instruction qui doit figurer dans la boucle **for** afin que la variable **Min** contienne en permanence le minimum des **i** premiers éléments du tableau.

---

**Question 5 - Modifications plus complexes****Question 5 - A : Les boutons "Permutation1" et "Permutation2"**

Un clic sur permutation1, permute circulairement les éléments du tableau "vers la droite". c.a.d remplace respectivement **N[2]** par **N[1]**, **N[3]** par **N[2]**, ..., **N[M]** par **N[M-1]** et **N[1]** par **N[M]**.

**Permutation2** agit dans le sens inverse.

Procédures appelées: **Permutation1** et **Permutation2**.

**Exemple (avec M = 4)**Le tableau avant permutation

i	1	2	3	4
N[ i ]	5	-1	4	2

Le tableau après permutation1 (vers la droite)

i	1	2	3	4
N[ i ]	2	5	-1	4

Le tableau avant permutation2 (vers la gauche)

i	1	2	3	4
N[ i ]	-1	4	2	5

**Indications (pour la permutation1 = vers la droite)**

Il faut commencer par la fin du tableau. Pour cela, utilisez une boucle **for** avec **downto** (à la place de **To**). Cette variante de la boucle **for** permet de décrémenter le compteur au lieu de l'incrémenter.

Il sera également nécessaire de sauvegarder la valeur du dernier élément avant d'entrer dans la boucle pour ne pas la perdre.

### Question 5 - B : Le bouton "Inverser"

Un clic sur ce bouton inverse l'ordre des éléments du tableau.

Procédure appelée: **Inverser**.

#### Exemple (avec M=4)

Le tableau avant inversion

i	1	2	3	4
N[ i ]	5	-1	4	2

Le tableau après inversion

i	1	2	3	4
N[ i ]	2	4	-1	5

**Indication:** inverser un tableau revient à échanger les éléments symétriques par rapport au milieu. Le problème est de trouver l'indice du symétrique du  $i$ ème élément en fonction de  $i$ . La fonction **floor** (partie entière) pourra vous être utile ici.

### Question 5 - C : Les boutons "Tri croissant" et "Tri décroissant"

Ces deux boutons permettent respectivement de ranger et les éléments du tableau dans l'ordre croissant et décroissant.

Procédure appelée: **TriCroissant** et **TriDeCroissant**.

Indication (pour le tri croissant)

Utilisez une boucle **for** avec un compteur  $i$  variant de 1 à ?.

Le problème est de trouver les instructions qui doivent figurer dans la boucle **for** afin que tableau soit toujours trié jusqu'au  $i$  élément.

On pourra utiliser une fonction (à compléter également) **IndiceDuMin( i )** qui retourne l'indice du plus petit éléments du tableau parmi les éléments d'indices  $i$  à  $M$ . En fonction de cet indice, on décidera d'échanger ou de ne pas échanger le  $i$ ème élément avec le plus petit élément qui le suit.

Exemple de déroulement de la boucle for

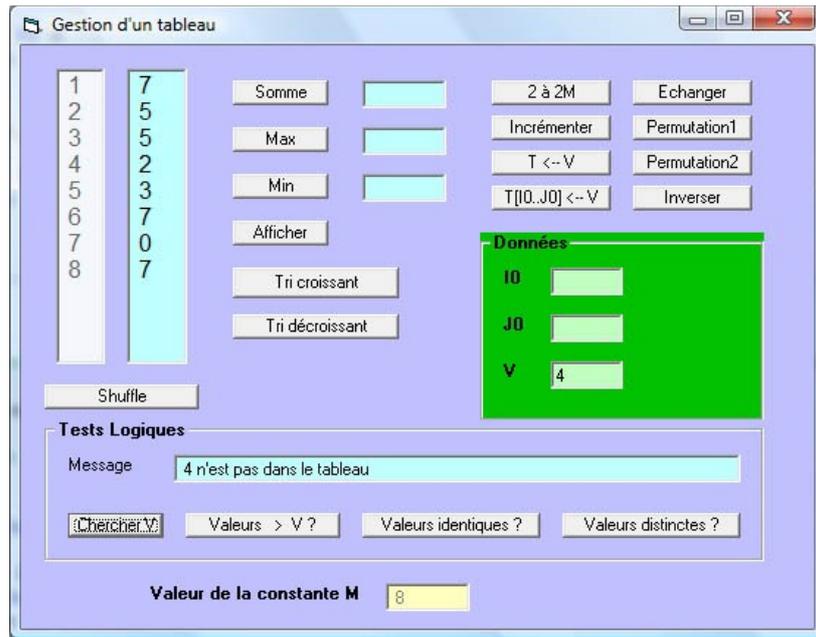
i	1	2	3	4	IndiceDuMin ( i )	Echanger
1	5	-1	4	2	2	1 et 2
2	-1	5	4	2	4	2 et 4
3	-1	2	4	5	3	Non
4	-1	2	4	5	4	Non

## Question 6 - Tests logiques

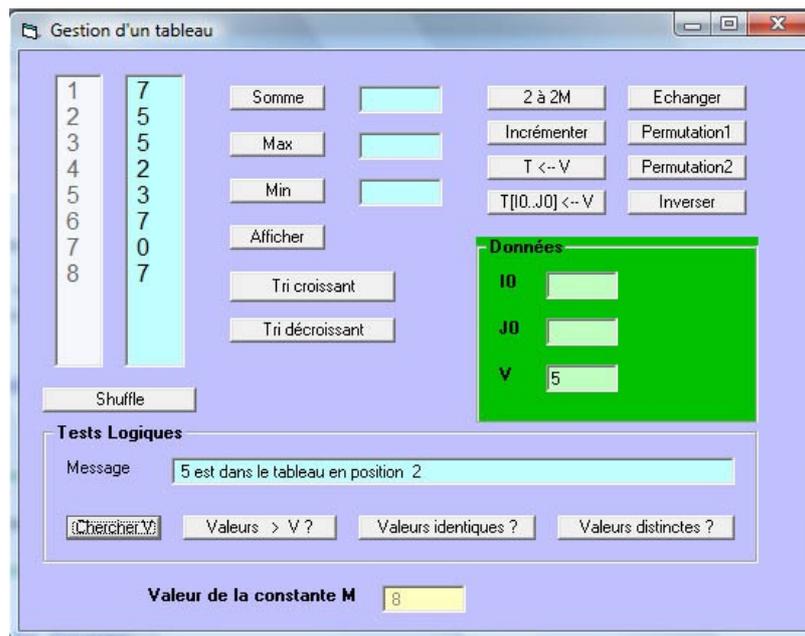
### Question 6 - A : Le bouton "Chercher V"

Lorsque l'utilisateur clique sur ce bouton, le programme doit lui indiquer si la valeur **V** se trouve dans le tableau et si oui, en quelle position. Procédure à compléter: **ChercherValeur**.

#### Exemple1 : Recherche de 4 - Echec de la recherche - Message : "4 n'est pas dans le tableau"



#### Exemple2: Recherche de 5 - La valeur est trouvée - Message : "5 est dans le tableau en position 2"



Remarque: la valeur **V** peut se retrouver plusieurs fois dans le tableau. Dans ce cas, on indiquera

simplement la première position (le plus petit indice).

### Indications

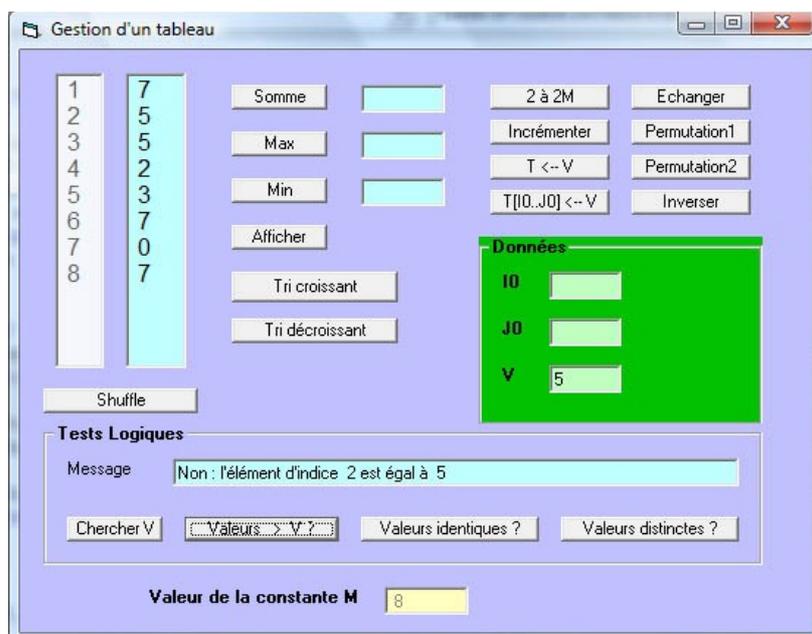
- Commencez simplement par détecter la présence ou l'absence de la valeur sans indiquer sa position.
- Utiliser un booléen **Trouve** que vous initialiserez à **False**.
- Parcourir les éléments du tableau avec une boucle **for**. A la sortie de cette boucle on doit avoir **Trouve = True** si et seulement si la valeur **V** a été rencontrée au moins une fois.
- Lorsque la détection de présence de la valeur fonctionne améliorez le message en indiquant la première position de la valeur lorsqu'elle est trouvée.

### Question 6 - B : Le bouton "Valeurs > V ?"

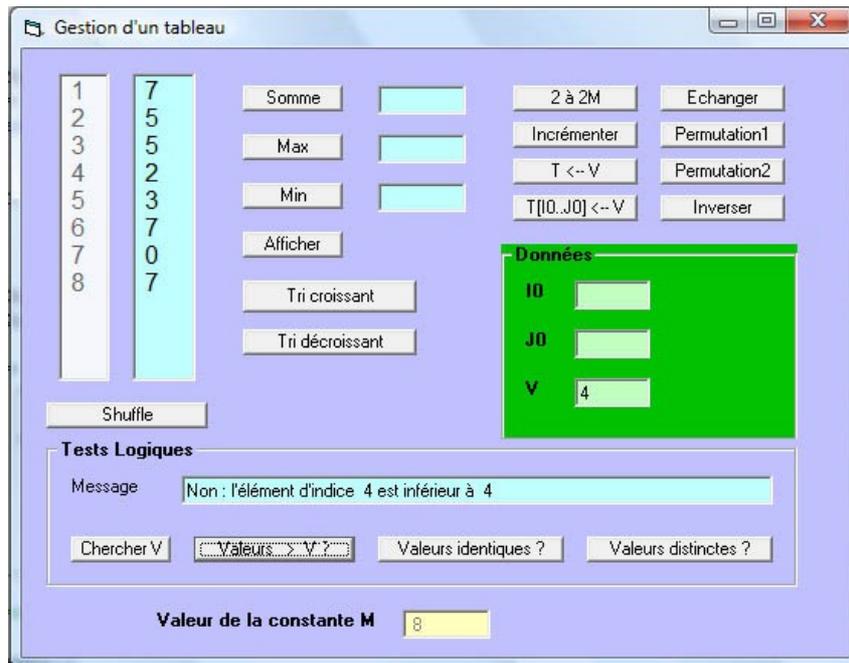
Lorsque l'utilisateur clique sur ce bouton, le programme doit lui indiquer si toutes les valeurs du tableau sont strictement supérieures à **V**. De plus, si cette propriété n'est pas vérifiée, le programme indique pourquoi.

Procédure à compléter: **TesterSiToutesLesValeursSontSuperieuresaV**.

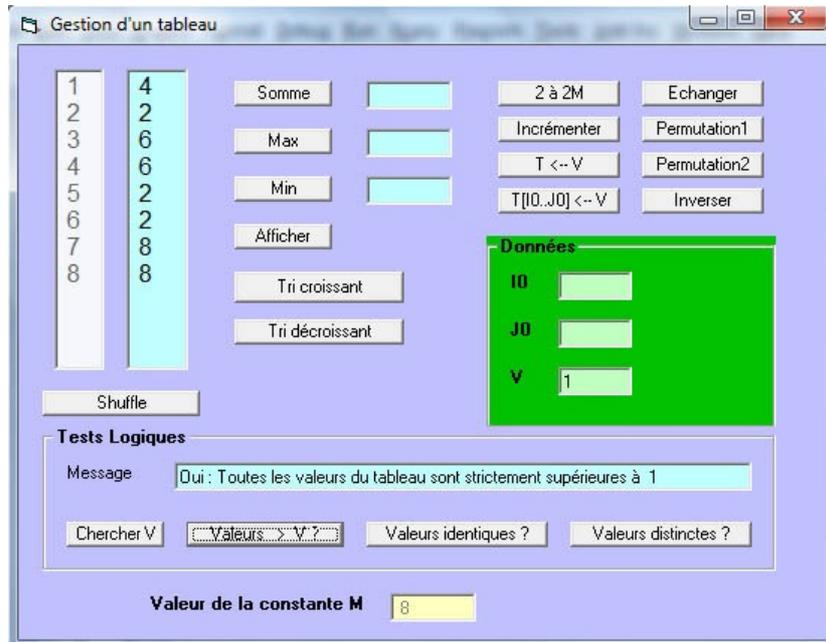
**Exemple 1: V = 5 - Message : "Non l'élément d'indice 2 est égal à 5"**



**Exemple 2:  $V = 4$  - Message : "Non : l'élément d'indice 4 est inférieur à 4"**



**Exemple 3:  $V = 1$  - Message : "Oui : Toutes les valeurs du tableau sont strictement supérieures à 1"**



**Question 6 - C : Le bouton "Valeurs identiques ?"**

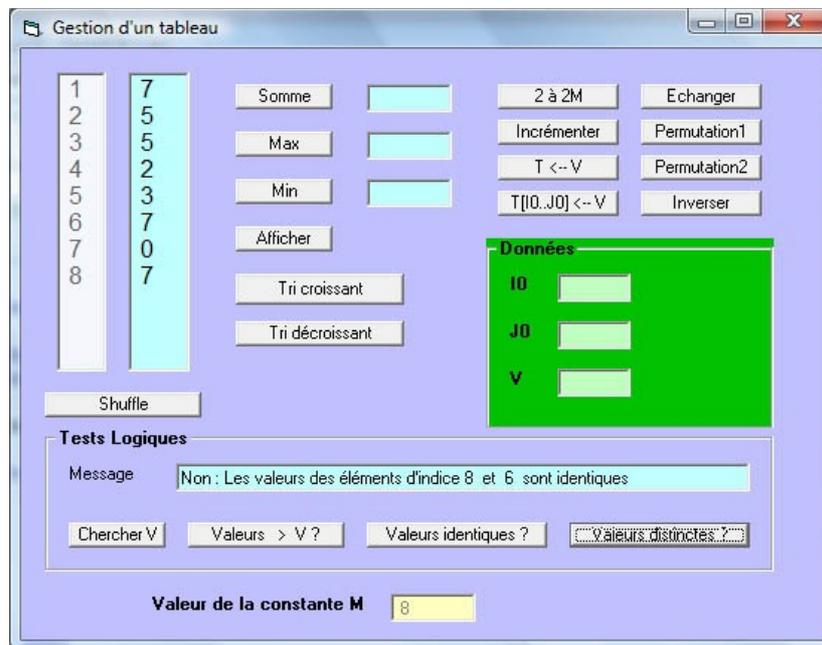
Lorsque l'utilisateur clique sur ce bouton, le programme doit lui indiquer si toutes les valeurs du tableau sont identiques.

Procédure à compléter: **TesterSiToutesLesValeursSontIdentiques**.

**Question 6 - D : Le bouton "Valeurs distinctes ?"**

Lorsque l'utilisateur clique sur ce bouton, le programme doit lui indiquer si toutes les valeurs du tableau sont distinctes. De plus, si cette propriété n'est pas vérifiée, le programme explique pourquoi (voir exemples ci-dessous).

Procédure à compléter: **TesterSiToutesLesValeursSontDistinctes**.

**Exemple: Message : "Non : Les valeurs des éléments d'indice 8 et 6 sont identiques"****Indications**

- Utilisez un booléen **ValDistinctes** que vous initialisez à **True**
- Avec deux boucles imbriquées comparez l'élément d'indice **i** avec l'élément d'indice **j**.
- Changer la valeur du booléen à **False** dès que vous rencontrez deux éléments identiques.

**Question 6 - E : Amélioration des performances**

Vous pouvez améliorer les procédures des questions A, B, C et D en arrêtant les itérations dès que l'on connaît la réponse à la question posée.

**Indication:** Utilisez pour cela des boucles **while** et des booléens.

**Question 7 - Modification de la dimension du tableau**

Faite fonctionner tout les boutons du projet **Gestab** avec un tableau de 11 éléments.