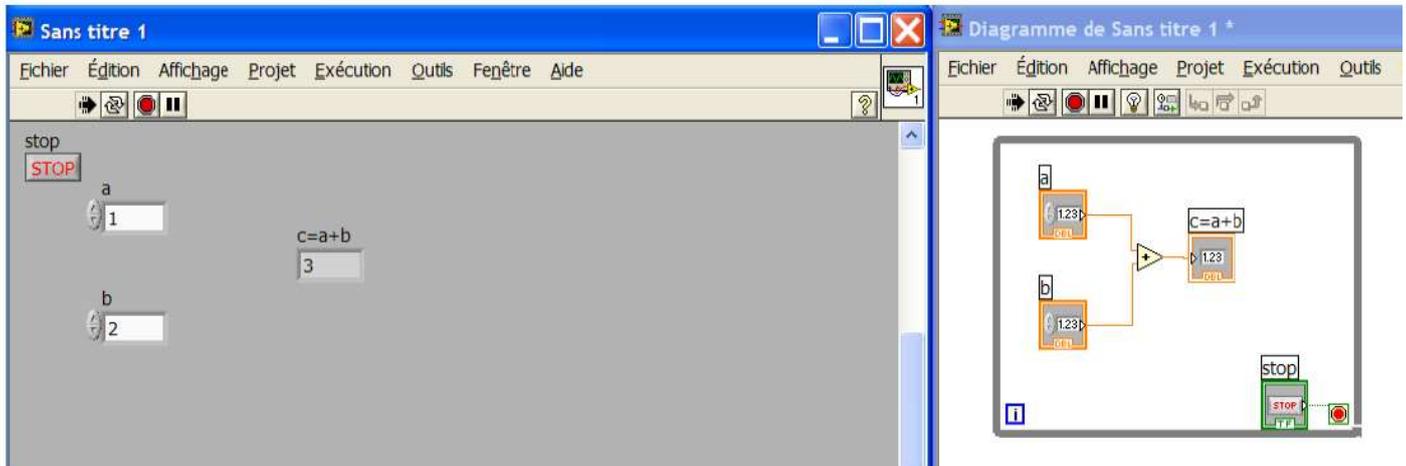


## 1 – Exemple 1 : Calcul simple

1.1 – Réaliser l'exemple ci-dessous (exemple 4a) du document ressource N°1). Faire constater le fonctionnement par le prof.

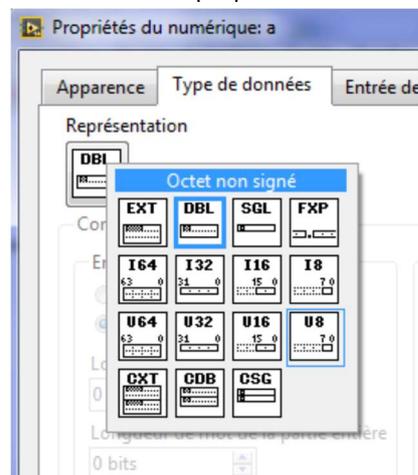


NB : Un nombre au format double est codé sur 64 bits (bit de signe – 11 bits d'exposant – 52 bits de mantisse)

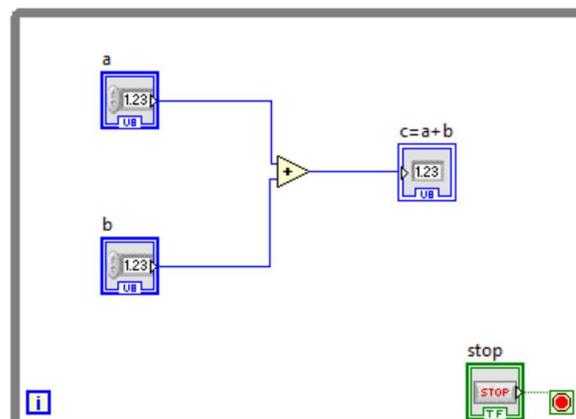
- ⇒ Vérifier que le programme fonctionne avec des grands nombres (faire un essai avec un nombre à 6 chiffres)
- ⇒ Vérifier que le programme fonctionne avec des nombres négatifs
- ⇒ Vérifier que le programme fonctionne avec des nombres à virgule

⇒ Modifier le format des données de a, b et c en U8 (octet non signé sur 8 bits), en éditant les propriétés :

Édition des propriétés de a



Pour obtenir le diagramme ci-dessous (les DBL oranges passent en U8 bleu) :



⇒ Tester avec a=5 et b=7

- ⇒ Tester avec a=245 et b=15
- ⇒ Essayer avec un nombre négatif
- ⇒ Essayer avec un nombre à virgule

1.2 - Justifier les résultats obtenus (en précisant les limites du format U8)

⇒ Modifier le format des données de a, b et c en i8 (octet signé sur 8 bits)

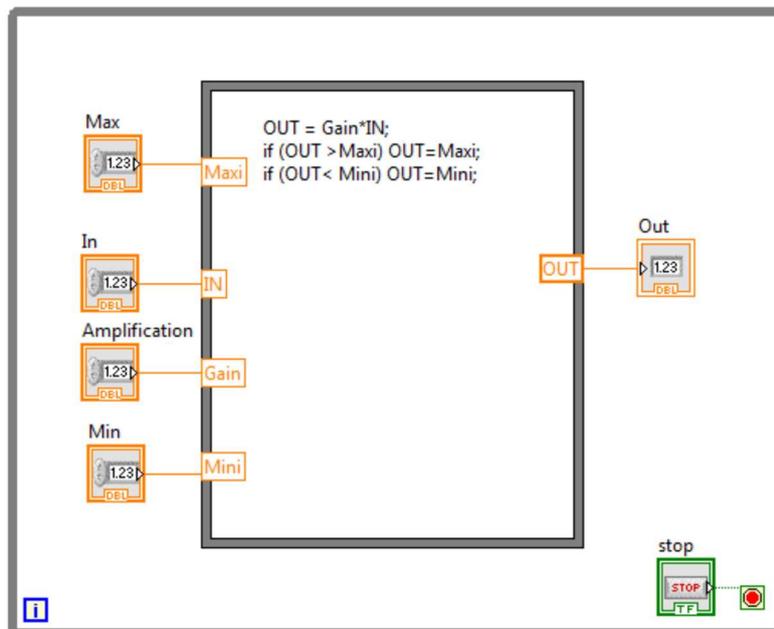
- ⇒ Tester avec a=5 et b=7
- ⇒ Tester avec a=-5 et b=15
- ⇒ Tester avec a=5 et b=126
- ⇒ Essayer avec un nombre à virgule

1.3 - Justifier les résultats obtenus (en précisant les limites du format I8)

## 2 – Exemple 2 : Boîte de calcul

La boîte de calcul permet de réaliser des fonctions logicielles écrites en langage C.

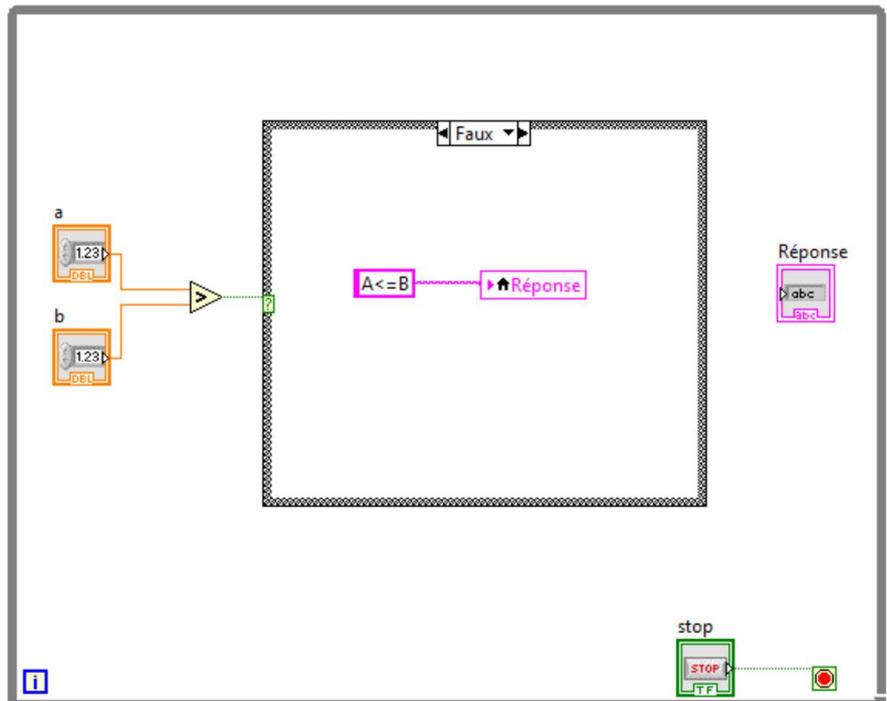
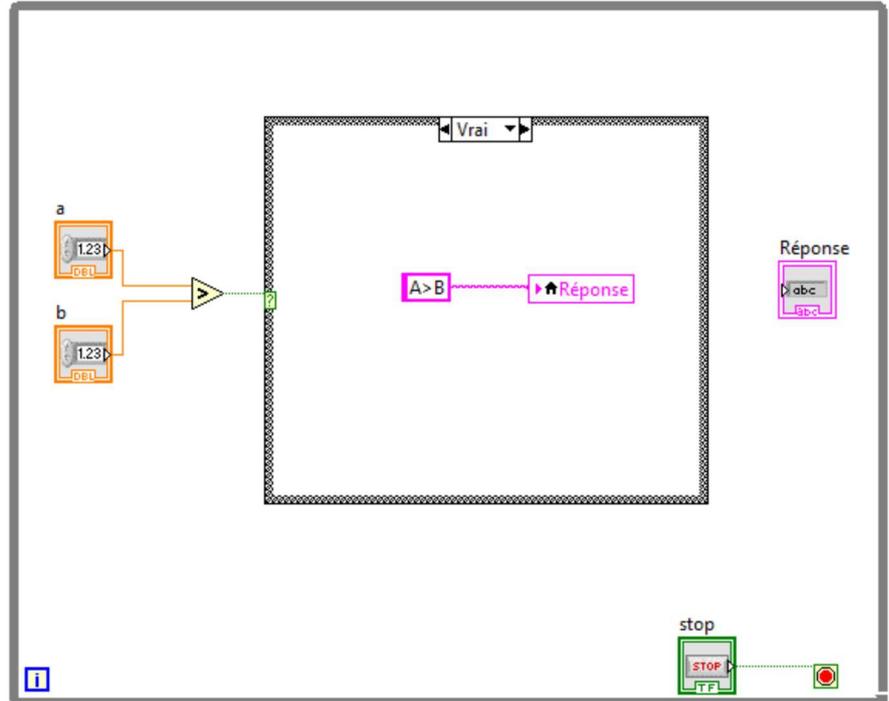
⇒ En s’aidant de l’exemple 4b) du document ressource N°1, réaliser le nouveau VI ci-dessous.



3.1 – Faire constater le fonctionnement au prof, et expliquer oralement ce que fait le programme

### 3 – Exemple 3 : Structure condition

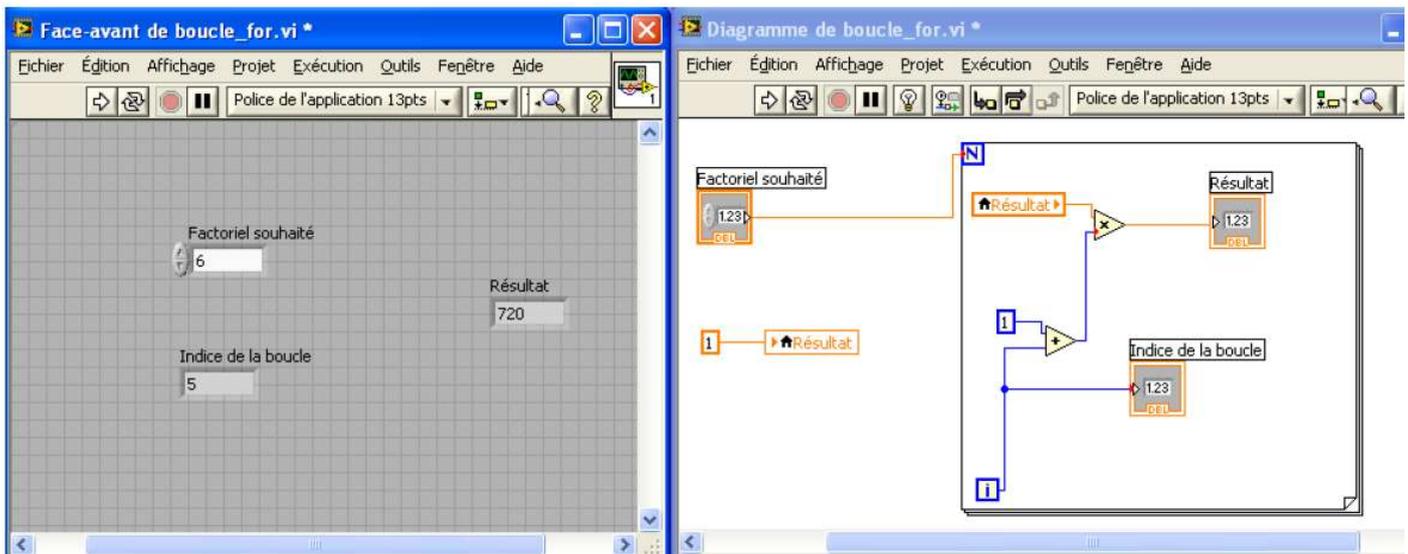
⇒ Réaliser l'exemple ci-dessous, qui est décrit dans le 1a) du document ressource N°2



3.1 – Faire constater le bon fonctionnement au prof et expliquer le fonctionnement du programme oralement.

## 4 – Exemple 4 : Boucle For

⇒ Réaliser l'exemple 1b) du document ressource N°2.

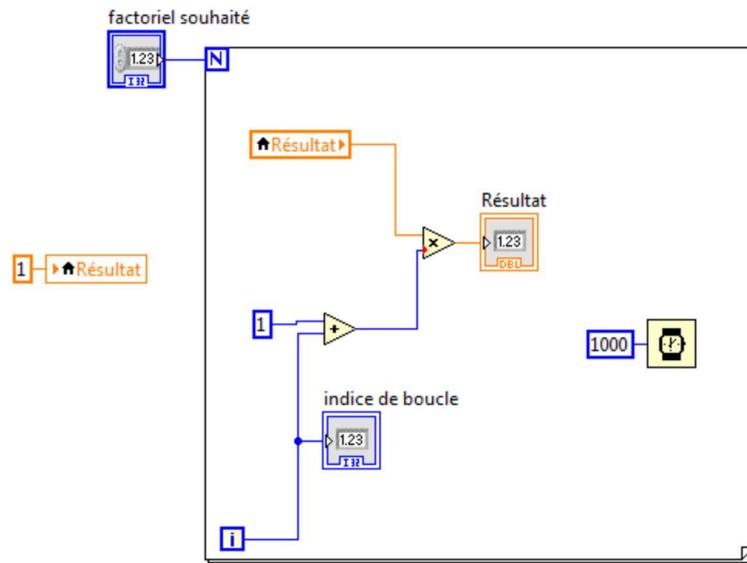


Remarques : Le factoriel d'un nombre  $n$  (noté  $n!$ ) est :  $1 \times 2 \times 3 \times 4 \times \dots \times n$

$$3! = 1 \times 2 \times 3 = 6$$

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$$

⇒ Ajouter un délai d'attente de 1 seconde (1000ms) entre chaque boucle, comme ci-dessous :

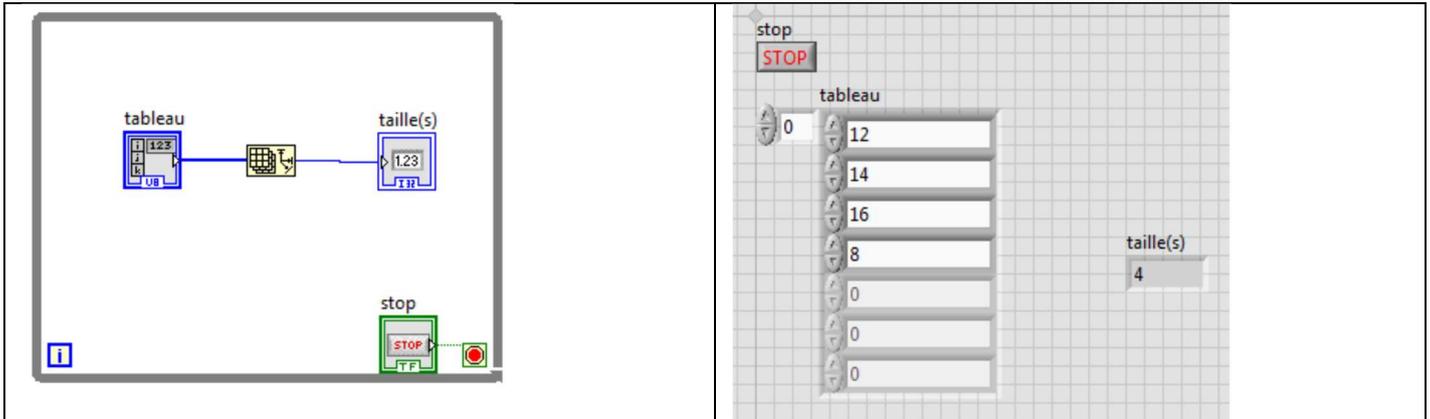


4.1 – Faire constater le bon fonctionnement au prof et expliquer oralement le fonctionnement de la boucle

## 5 – Les tableaux

### 5.1 – Taille d'un tableau

⇒ Ouvrir le fichier Tableau\_1.vi sous LABVIEW



Ici est déclaré un tableau d'octets (8 bits) non signés : U8

Ici 4 valeurs du tableau ont été initialisées (les parties grisées ne font pas partie du tableau). La taille du tableau est donc de 4 éléments.

Chaque élément comporte un indice, qui commence toujours par 0.

Dans notre exemple :

- l'élément d'indice 0 vaut 12, on notera `tableau[0]=12`
- l'élément d'indice 1 vaut 14, on notera `tableau[1]=14`
- l'élément d'indice 2 vaut 16, on notera `tableau[2]=16`
- l'élément d'indice 3 vaut 8, on notera `tableau[3]=8`

La première valeur affichée dans le tableau est celle dont l'indice est noté à gauche du tableau (ici 0).

⇒ Exécuter le programme

⇒ Augmenter l'indice du tableau avec la flèche du haut

⇒ Ajouter 2 éléments au tableau et indiquer la nouvelle taille du tableau

⇒ Faire une démonstration au prof et expliquer oralement

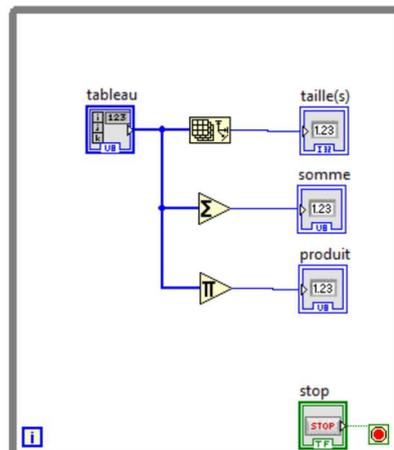
### 5.2 - Somme et multiplication des éléments d'un tableau

⇒ Réaliser le vi ci-dessous (Nouveautés : Somme et Produit des éléments d'un tableau)

⇒ Faire un essai avec un tableau de taille 4

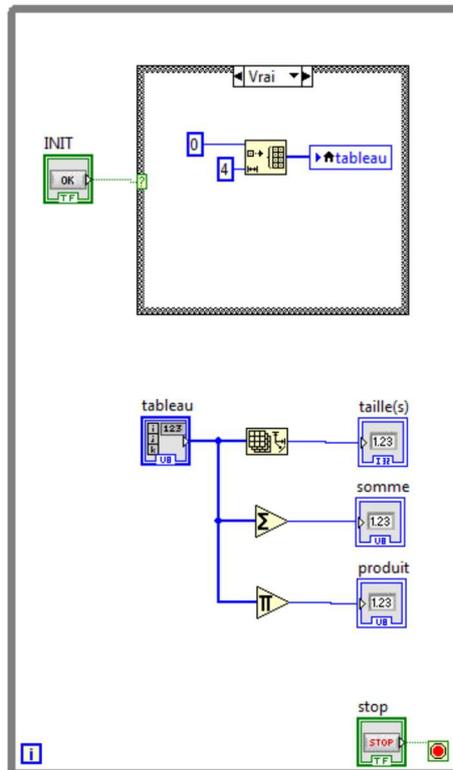
⇒ Vérifier que la limite supérieure de somme et produit est 255.

⇒ Faire une démonstration au prof. Expliquer.



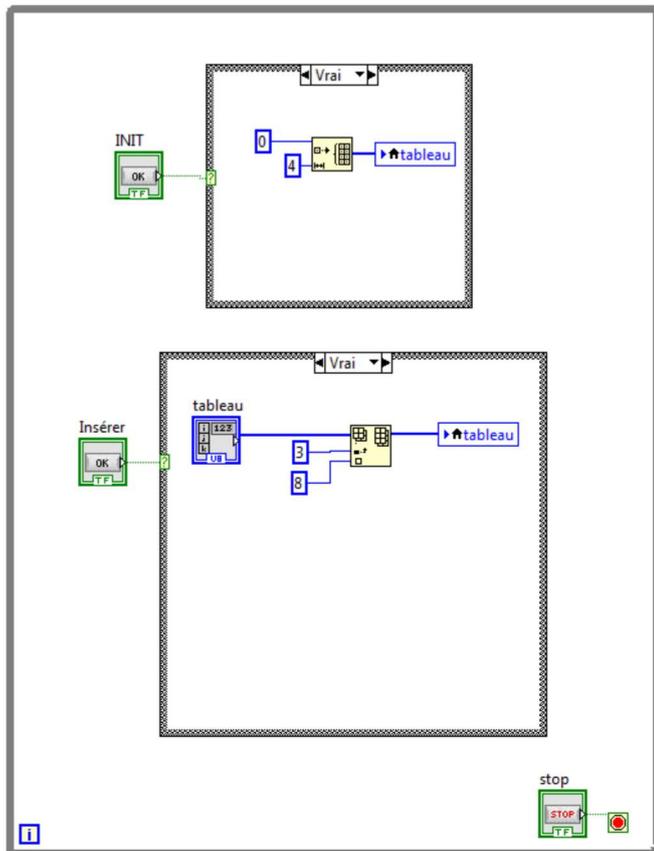
### 5.3 – Initialisation d'un tableau

- ⇒ Réaliser le vi ci-dessous (Nouveautés : Initialiser un tableau – Boucle condition – Variable locale – Bouton OK)
- ⇒ Faire un essai
- ⇒ Proposer une modification pour initialiser le tableau à une taille de 5
- ⇒ Proposer une modification pour initialiser le tableau avec la valeur 1
- ⇒ Faire une démonstration au prof. Expliquer.



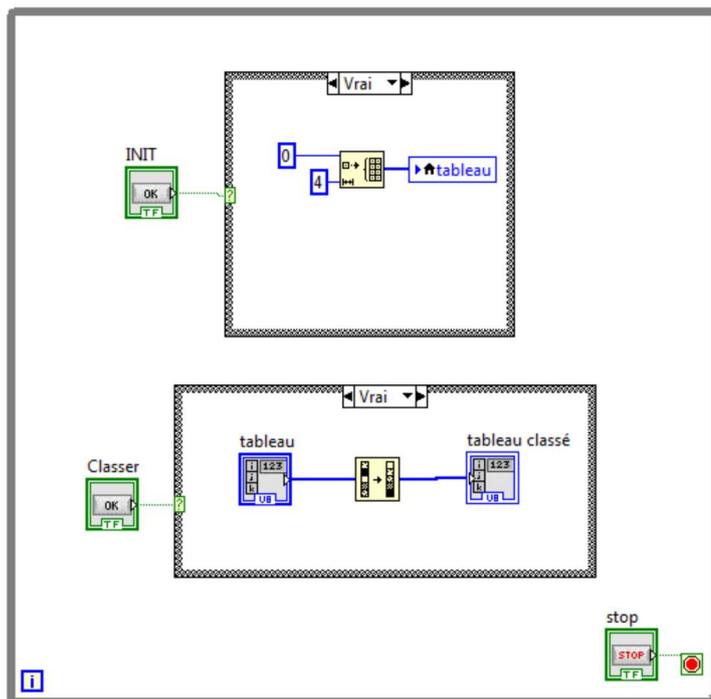
## 5.4 – Insertion d'un élément dans un tableau

- ⇒ Réaliser le vi ci-dessous (Nouveautés : Insérer dans un tableau)
- ⇒ Faire un essai
- ⇒ Proposer une modification pour insérer la valeur 5 dans le tableau à l'indice 1
- ⇒ Déterminer le vi à utiliser pour remplacer un élément, et non plus insérer un élément.
- ⇒ Faire une démonstration au prof. Expliquer.

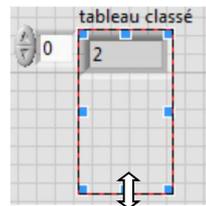


### 5.5 – Classer les éléments d'un tableau

- ⇒ Réaliser le vi ci-dessous (Nouveautés : Classer un tableau 1D)
- ⇒ Faire un essai
- ⇒ Faire une démonstration au prof. Expliquer.

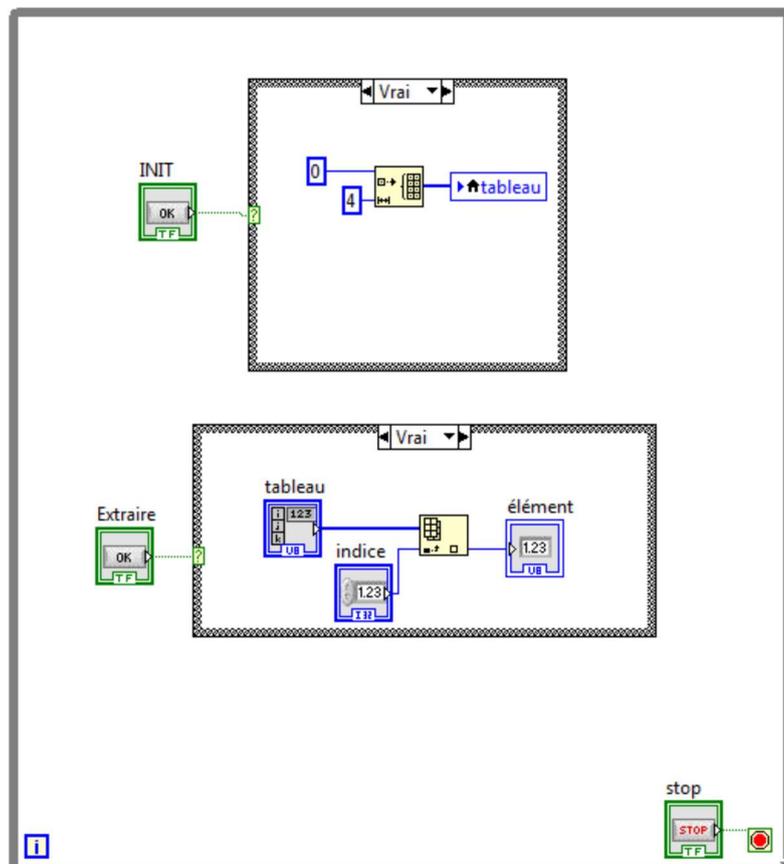


Note : Pour afficher plusieurs valeurs du tableau, il faut étirer la zone vers le bas avec la souris



### 5.6 – Lire la valeur d'un élément d'un tableau

- ⇒ Réaliser le vi ci-dessous (Nouveautés : Indexer un tableau)
- ⇒ Faire un essai (à chaque appui sur Extraire, la fonction renvoie l'élément du tableau dont l'indice est précisé en entrée)
- ⇒ Faire une démonstration au prof. Expliquer.

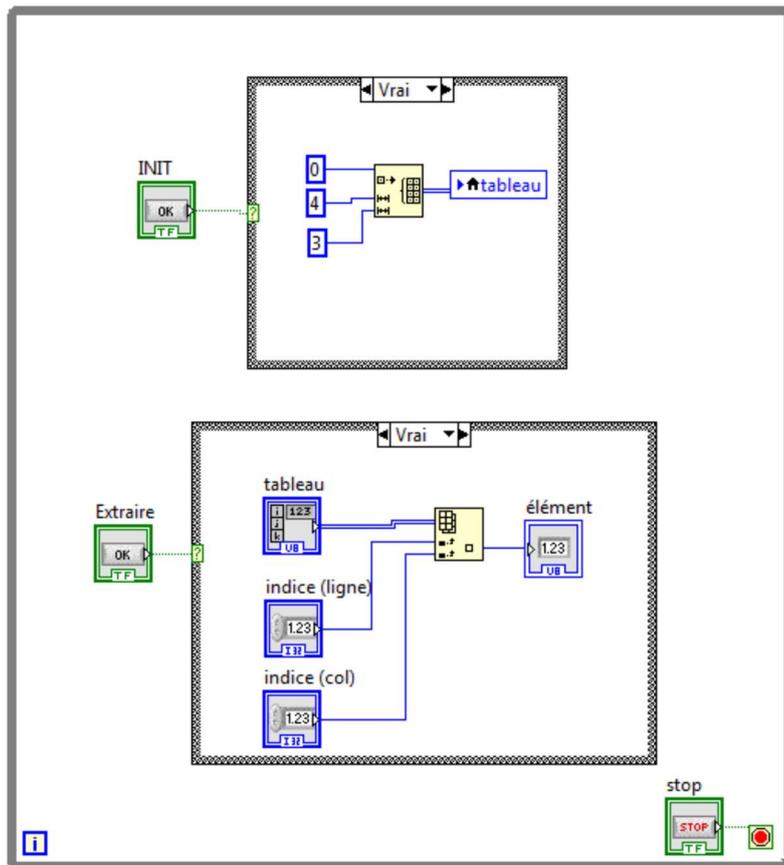


## 6 – Tableau de dimension 2 (ou plus)

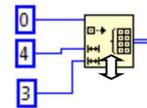
Un tableau peut avoir plusieurs dimensions.

Dans le vi ci-dessous, un tableau à 2 dimensions est déclaré (4 lignes – 3 colonnes).

⇒ Réaliser le vi ci-dessous dans labview.



Etirer vers le bas avec la souris pour ajouter une dimension



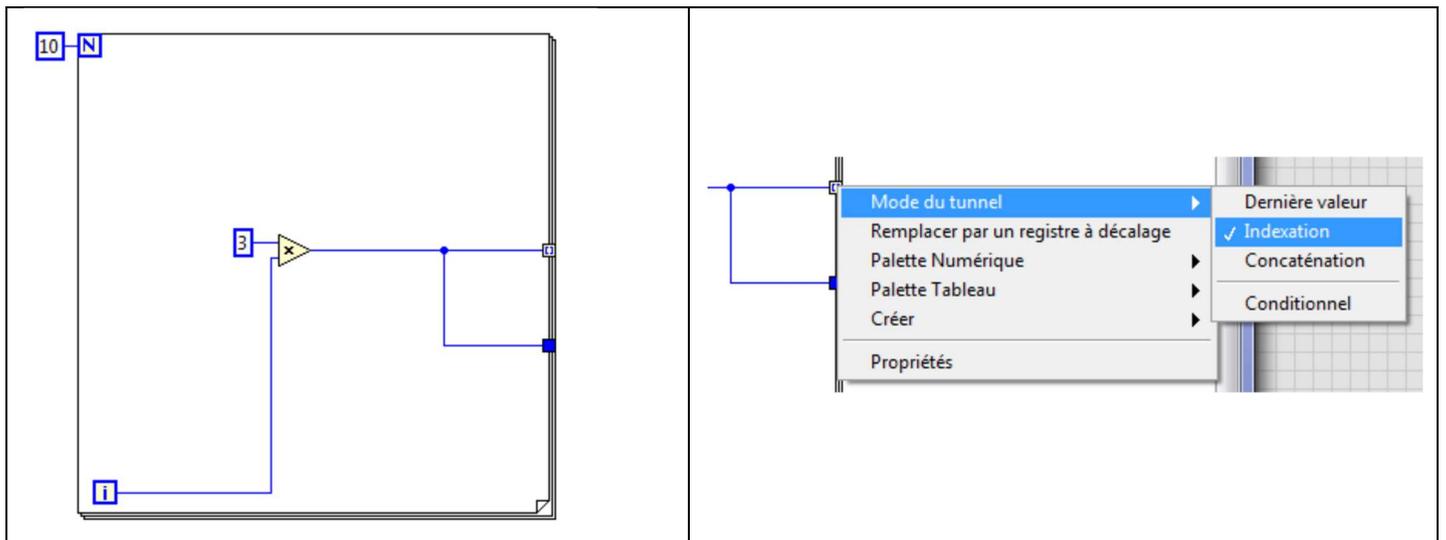
⇒ Tester le fonctionnement comme ci-dessous. Expliquer oralement les résultats obtenus.

tableau	indice (ligne)	indice (col)	élément
1	2	2	11
2			
3			
4			
0			
0			
0			
0			

NB : En langage C (ou C++), la déclaration d'un tableau de dimension 2, d'octets non signés se fait de la manière suivante :  
**unsigned char mon\_tableau[4][3];**

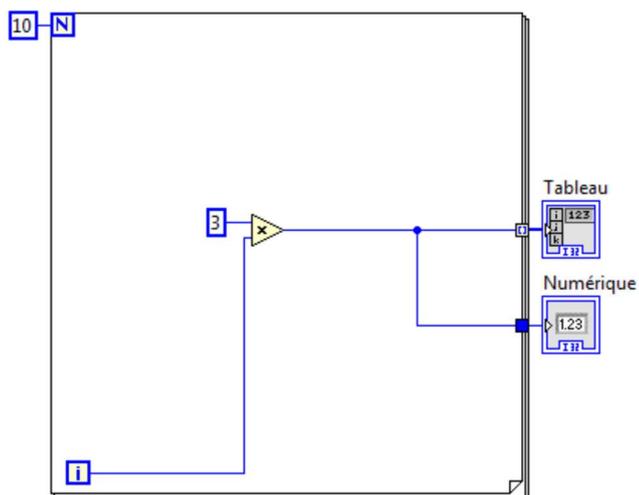
## 6 – Donnée renvoyé par une boucle

⇒ Réaliser le vi ci-dessous dans LABVIEW.



Une terminaison de boucle est paramétrée en Indexation (clic droit sur la terminaison), l'autre en Dernière valeur.

⇒ Créer des indicateurs sur les terminaisons comme ci-dessous



⇒ Exécuter le vi

⇒ Indiquer les valeurs renvoyées à la fin de la boucle FOR

⇒ Justifier les valeurs renvoyées

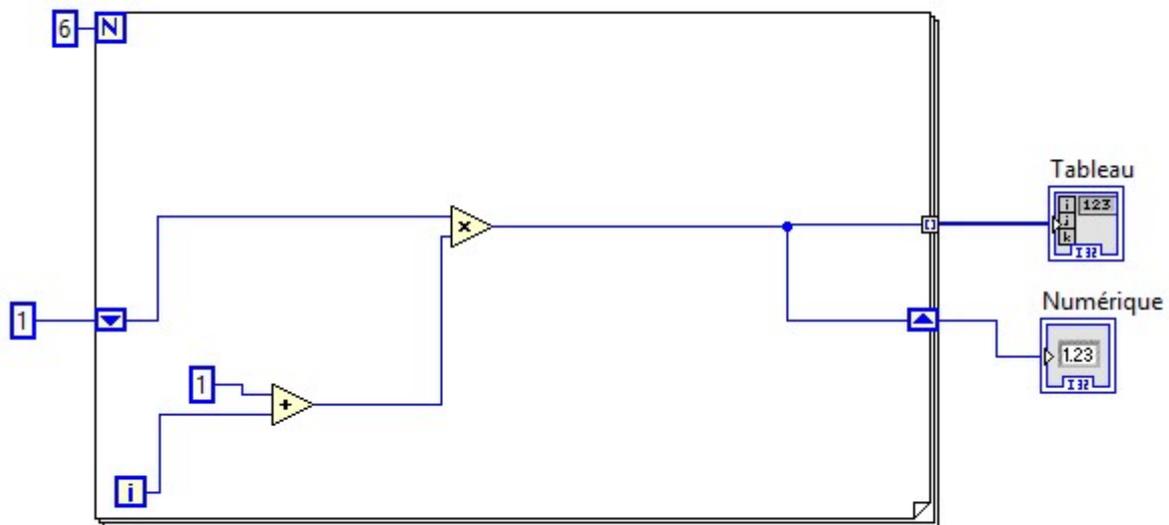
⇒ Justifier la taille du tableau

⇒ Modifier le vi pour avoir une taille de tableau égale à 15

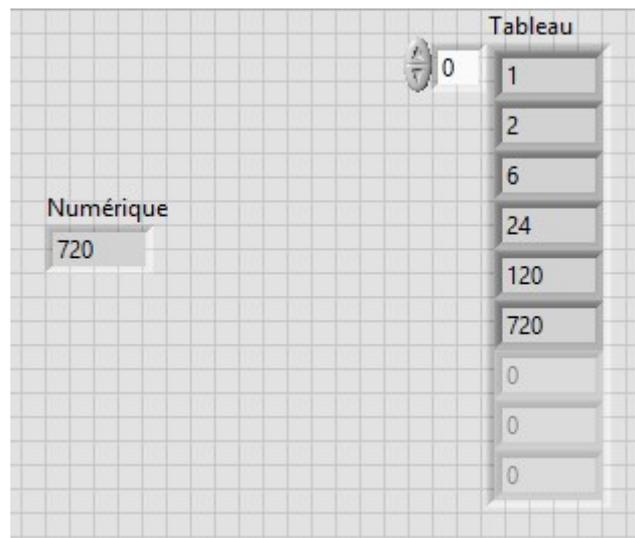
⇒ Appeler le prof pour donner les explications oralement.

### Registre à décalage :

⇒ Modifier le vi de la manière suivante (la sortie numérique est remplacée par un registre à décalage):



⇒ Exécuter le vi et visualiser le résultat comme ci-dessous :



Explication : La sortie  est renvoyée dans l'entrée  qui au départ est initialisée à 1 à la première exécution. Il est possible de mettre plusieurs registres à décalage. L'entrée et la sortie correspondante sont alignées horizontalement.

⇒ Indiquer la fonction réalisée et donner des explications orales au prof.