

- 2) 1) De quoi est composée une variable globale fonctionnelle ?  
(Expliquez le principe de fonctionnement)  
(Détail, Voir site NI)

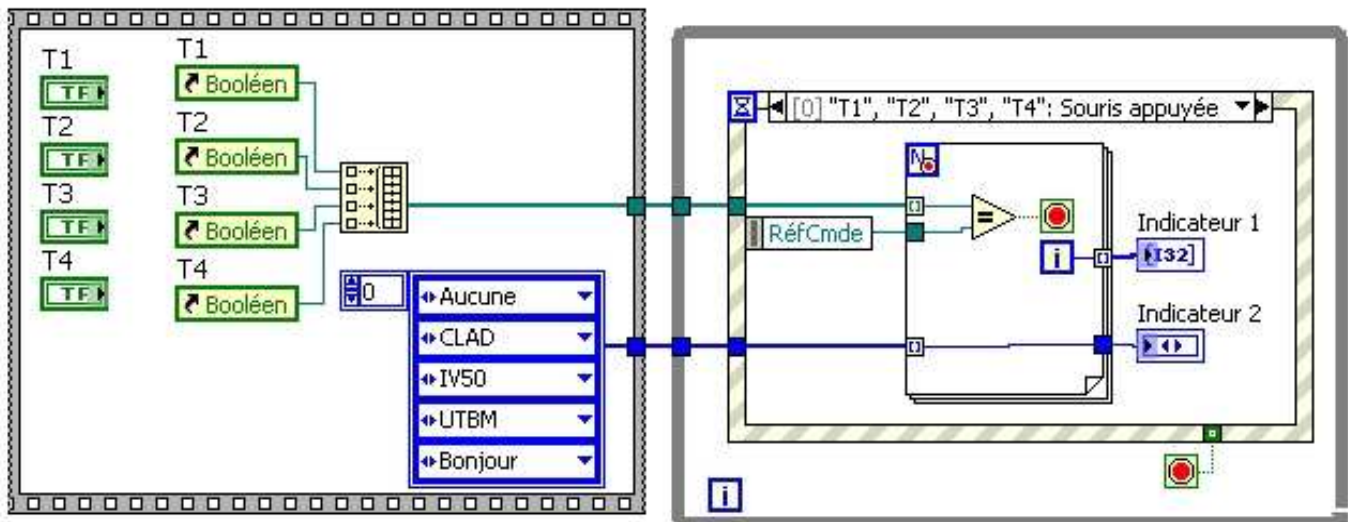
Le principe repose sur l'utilisation de registres à décalage et sur l'état de leur initialisation. Dans le cas d'une variable globale fonctionnelle, on utilise la propriété de mémorisation de la dernière valeur stockée dans le registre. Cette mémorisation est effectuée en utilisant une boucle While qui ne s'exécute qu'une seule fois grâce à une condition d'arrêt toujours valide. La valeur stockée dans la variable Résultat sera donc conservée pour tous les programmes qui vont utiliser la FGV.

On utilise une FGV par l'intermédiaire d'actions qui feront parti des paramètres d'entrée de la variable. En général, les actions les plus standards sont la lecture et écriture d'information. Mais les possibilités d'actions sont très flexibles.

- 2) 2) Comment une variable globale fonctionnelle protège-t-elle une action de lecture ou d'écriture contre les appels simultanés de plusieurs VIs ?

Le VI FGV (fonctionnal global variable) utilise la propriété par défaut d'exécution non-réentrante, c'est-à-dire que l'appel au sous-VI FGV devra terminer son exécution complètement avant que l'appel par un autre VI ne puisse être pris en compte et exécuté.

- 4) 3) Considérons le programme suivant :



Définir le type de donnée des indicateurs 1 et 2 :

**Indicateur 1 :**

Tableau à une dimension d'entier 32bits.

**Indicateur 2 :**

Type énumération.

Si on effectue un clic gauche de souris sur le bouton **T3**, donnez la valeur des indicateurs 1 et 2

La structure événement exécutera la boucle FOR jusqu'à ce que la condition de sortie de boucle soit vérifiée. C'est-à-dire jusqu'à ce que le refnum de la commande qui a déclenché l'événement corresponde à celui qui est indexé par la boucle (celui du bouton T3). La boucle s'exécutera donc 3 fois.

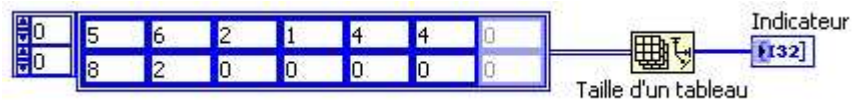
**Indicateur 1 :**

Tableau contenant les valeurs [0 ;1 ;2]

**Indicateur 2 :**

Valeur énum IV50

- 2 4) Dans le programme suivant donnez le type de donnée et la valeur de l'indicateur après exécution du programme.



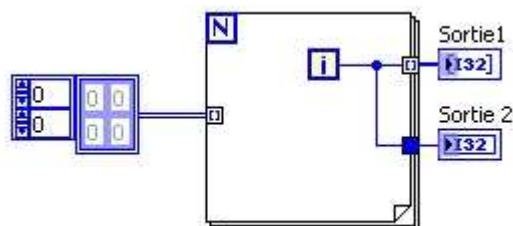
**Type de donnée :**

Tableau à une dimension d'entier 32bits.

**Valeur :**

[2 ;6] (car il y a 2 lignes et 6 colonnes).

- 2 5) Dans le programme suivant donnez la valeur des indicateurs sortie 1 et 2 après exécution du programme



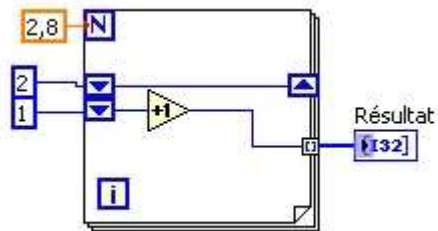
Attention piège !!!

La boucle FOR ne s'exécutera pas car le tableau d'entrée est vide.

La sortie 1 sera donc un tableau vide.

La sortie 2 prendra la valeur 0 car elle ne peut pas ne pas avoir de valeur (c'est un cas particulier).

3) 6) Considérons le diagramme suivant :



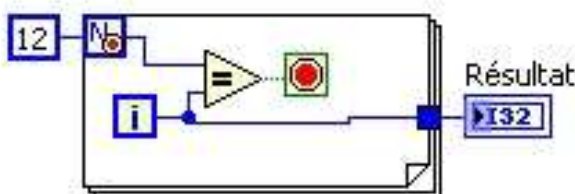
Combien de fois la boucle s'exécutera-t-elle ?

Le point de coercition indique que le réel (2,8) sera converti en entier (arrondi à 3). La boucle s'exécutera donc 3 fois.

Donnez la valeur de l'indicateur Résultat après exécution du programme.

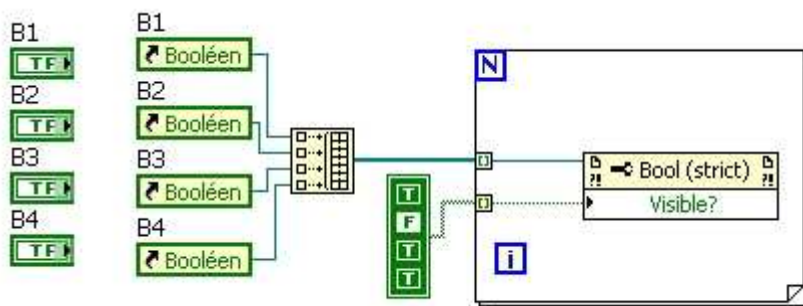
Résultat est le tableau d'entiers suivant : [2 ; 3 ; 3]

1) 7) Que vaut l'indicateur résultat après exécution du programme suivant :



La boucle s'exécutera 12 fois. Le terminal de condition ne sera pas utilisé pour sortir de la boucle car i varie de 0 à N-1. Le résultat sera donc l'entier 11.

2) 8) Expliquez ce que réalise le programme suivant :



Ce programme modifie la propriété « visible » des commandes booléennes B1 à B4. Plutôt que d'utiliser 4 nœuds de propriété implicites, on utilise ici un nœud explicite dans une boucle FOR qui indexe le tableau de refnum des commandes concernées. Après exécution, nous aurons :

- B1 visible
- B2 invisible
- B3 visible
- B4 visible

1) 9) Expliquez la différence entre un événement de type filtre et un événement de type notification.

(Détail, Voir aide labview)

Les événements de type notification indiquent qu'une action utilisateur a eu lieu et que LabView l'a traité (exp : appui sur un bouton -> le bouton est enfoncé).

Les événements de type filtre informent du fait que l'utilisateur a effectué une action avant que LabVIEW ne la traite, ce qui vous permet de personnaliser la façon dont le programme répond aux interactions avec l'interface utilisateur (exp : appui sur un bouton -> le bouton n'est pas encore enfoncé, l'utilisateur peut décider de rejeter l'action ou personnaliser la façon de la traiter).

2) 10) listez les différentes façons de transmettre des données :

**Entre plusieurs VI fonctionnant sur une même machine :**

Utilisation de :

- Variable globale
- Variable globale fonctionnelle
- File d'attente
- Notificateur
- Variable partagée
- Fichiers
- (éventuellement Communication réseau TCP, UDP, etc..)

**Entre plusieurs VI fonctionnant sur des machines différentes reliées sur un réseau Ethernet :**

Utilisation de :

- Variable partagée publiée sur le réseau
- Fichiers par service FTP
- (éventuellement Communication réseau TCP, UDP, etc..)

2) 11) Peut-on transmettre des données d'un VI à un autre sur la même machine en utilisant une file d'attente ? (expliquez pourquoi et éventuellement comment)

Oui, il suffit de transmettre le refnum de la file d'attente grâce à une variable globale.