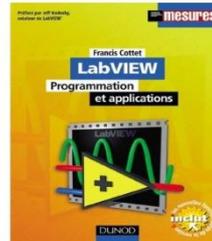
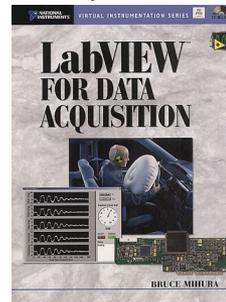


Bibliographie :

- ① Initiation à Labview, Edition d'Août 2003 référence 350784B-01- National Instruments
- ② LabVIEW : Programmation et applications, Francis Cottet



- ③ LabVIEW for Data Acquisition, Bruce Mihura



Sommaire

- ✓ Environnement LABVIEW
- ✓ Construction des VI et des sous VI
- ✓ Structures de contrôle
- ✓ Fonctionnalités avancées
- ✓ Accès au contenu de la mémoire
- ✓ Opération sur les fichiers

Objectifs du cours

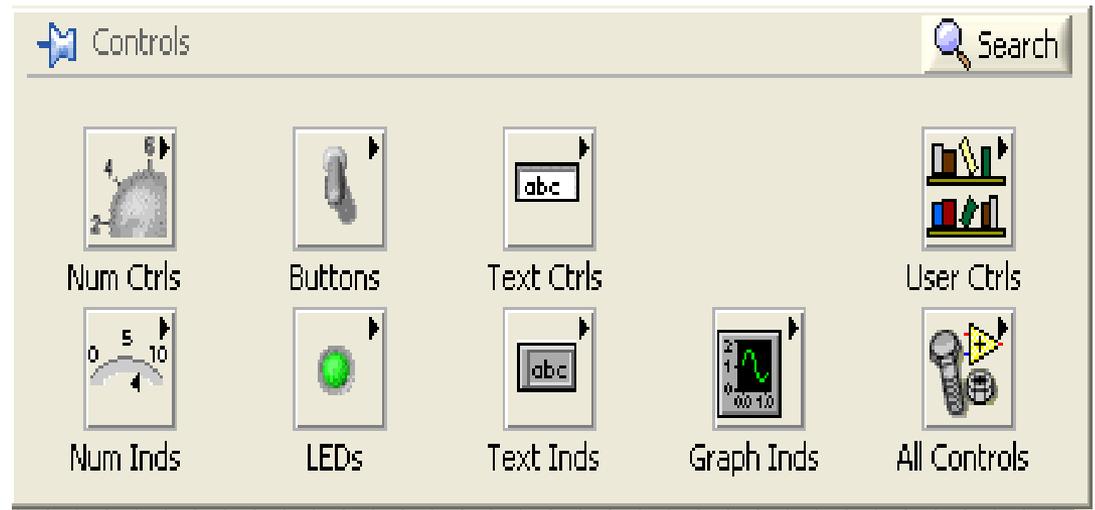
- Comprendre les faces-avant, les diagrammes, les icônes et les connecteurs*
- Utiliser les structures de programmation et les types de données qui existent dans LabVIEW*
- Utiliser différentes techniques d'édition et de mise au point*
- Créer et enregistrer des VIs pour les utiliser comme sous-Vis*
- Afficher et enregistrer des données*

Boite à outils



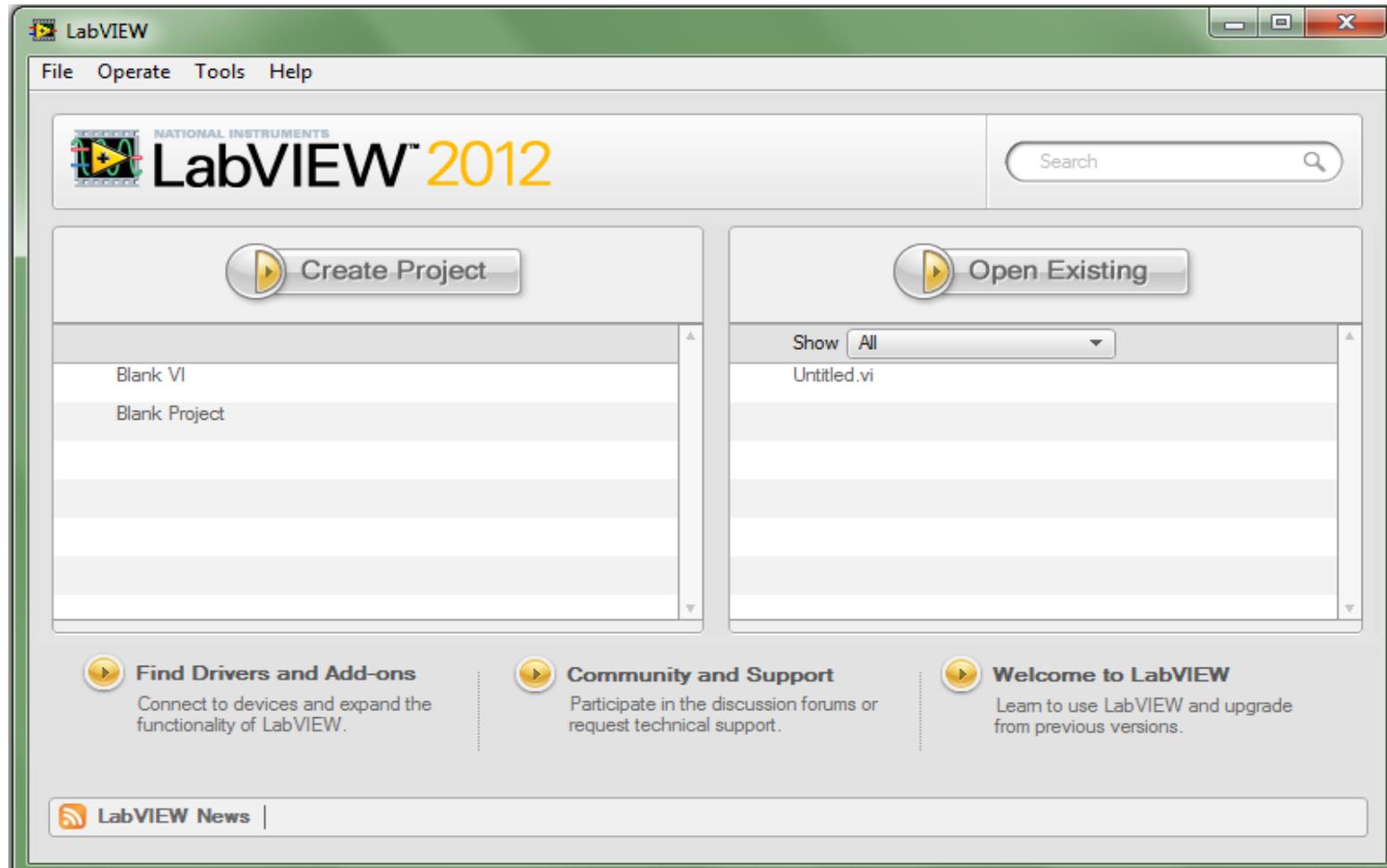
Boite à outils

*Instruments
Virtuels*



LABVIEW

Lancement de l'environnement LabVIEW : fenêtre de démarrage s'affiche



Terminologies

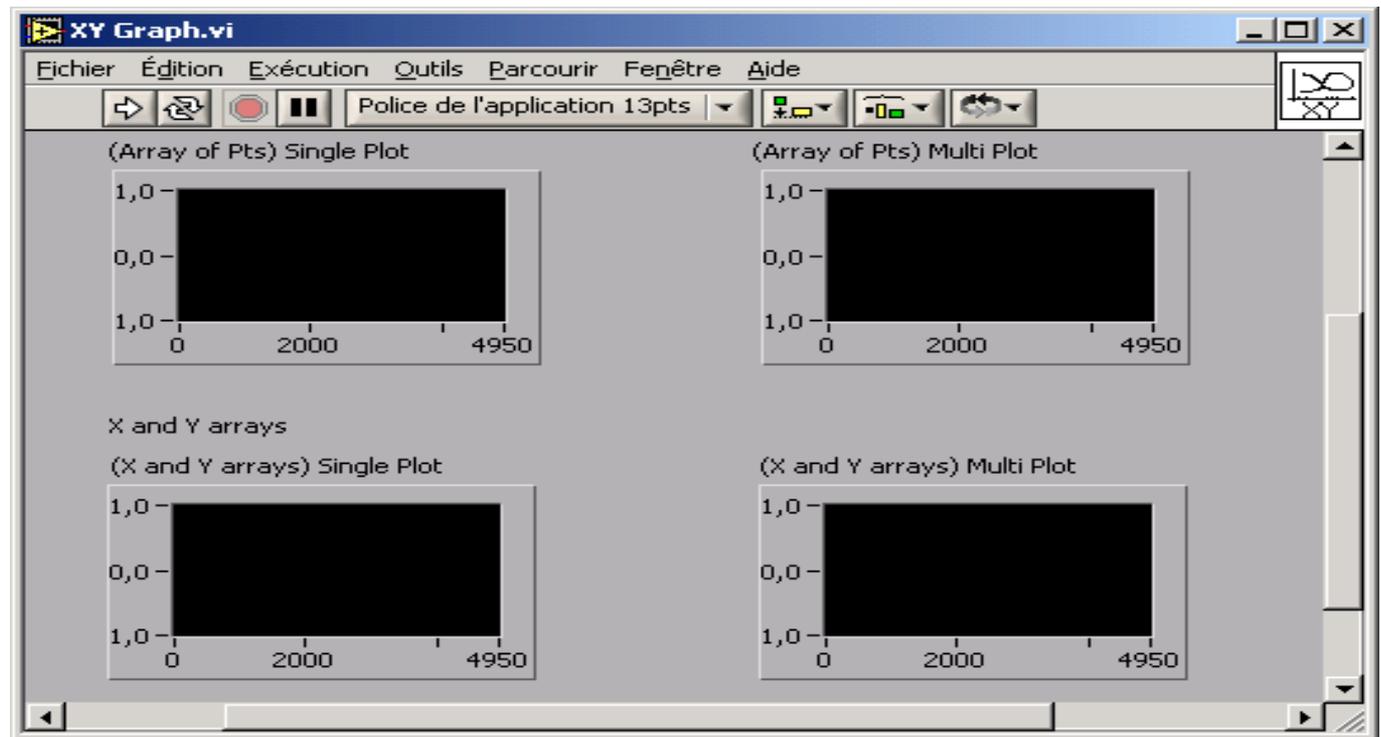
- ❑ Les VIs Labview se composent de trois composantes principales :
 - La fenêtre face avant (s'affiche automatiquement lorsque vous ouvrez un nouveau VI)
 - La fenêtre diagramme
 - les icones/connecteurs

- ❑ les objets de la fenêtre face avant s'appellent les terminaux
- ❑ les objets de la fenêtre diagramme qui comportent des entrées et/ou des sorties et qui réalisent des opérations pendant l'exécution du VI s'appellent les nœuds
- ❑ les VIs express s'appellent des nœuds extensibles

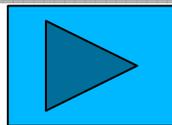
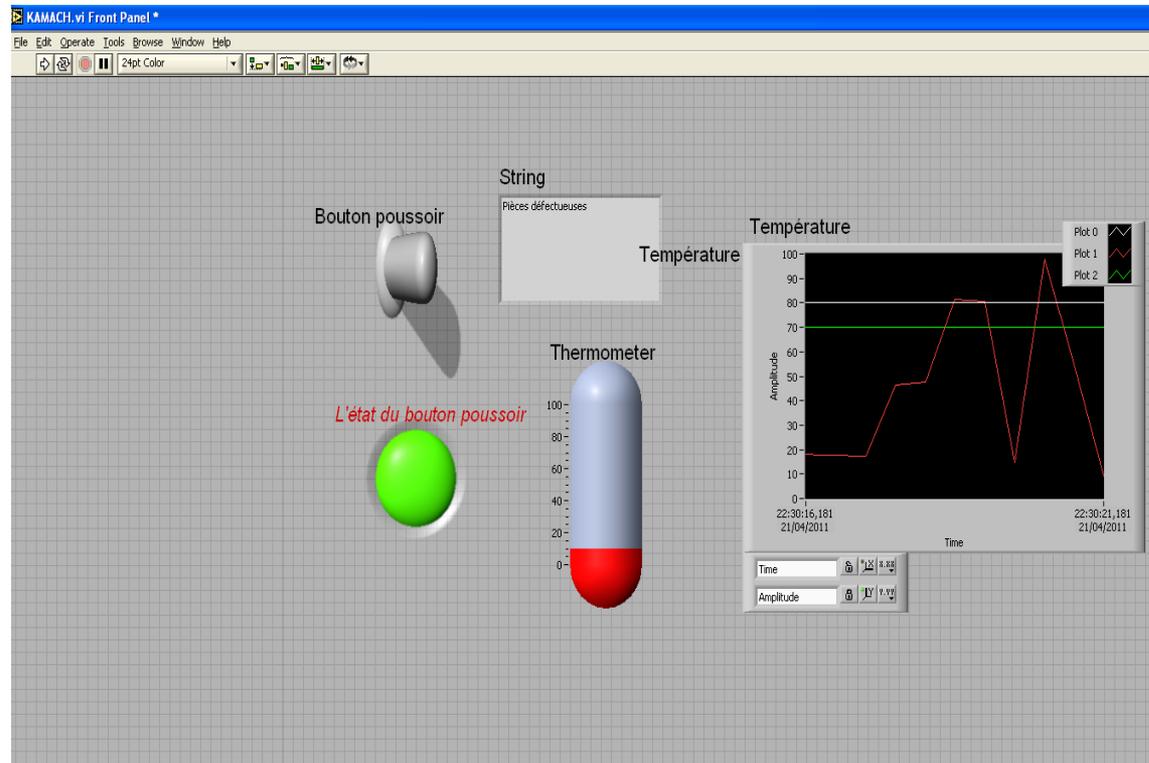
Fenêtres

☐ Fenêtre principale :

- ✓ face avant c'est l'interface avec l'utilisateur,
- ✓ chargée d'afficher les résultats de calcul sur l'écran de l'ordinateur



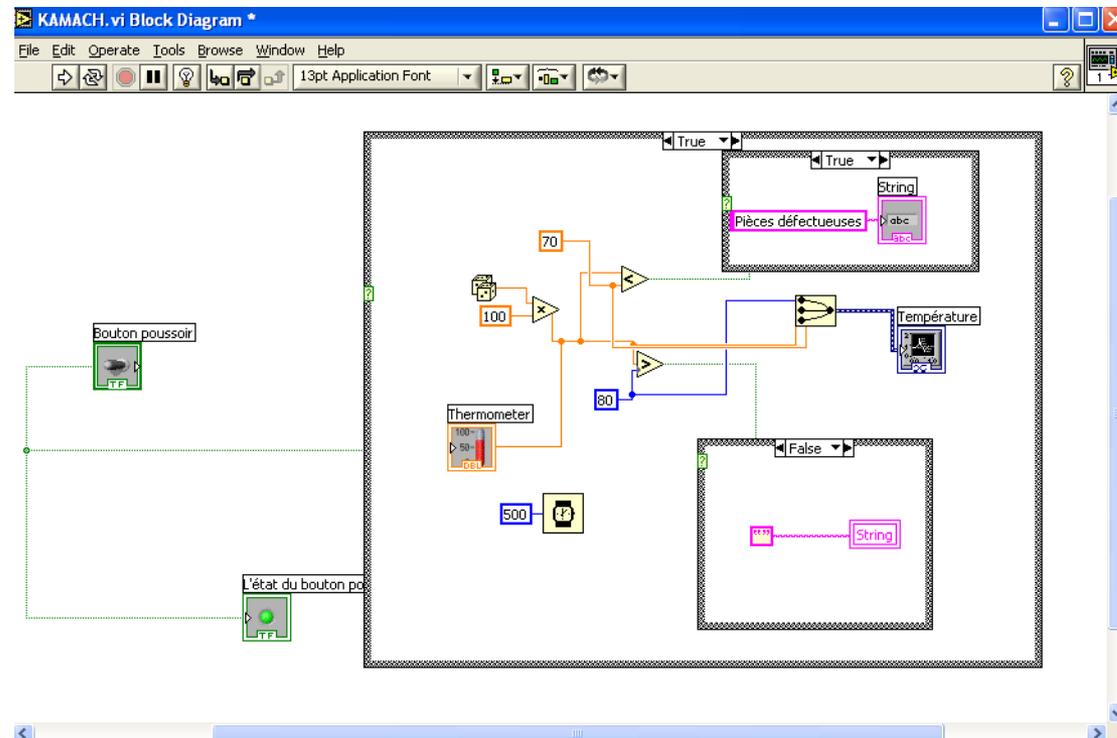
Construction de la fenêtrés : Face avant *Exemple pour afficher la température d'un four*



Fenêtres

❑ Fenêtre principale :

- ✓ diagramme, correspondant au programme
- ✓ Elle décrit le fonctionnement interne d'un VI



Mode d'exécution :

- ✓ Mode pas à pas : le bouton  lance l'exécution
(ou affiche la liste des erreurs en cas d'erreur)
- ✓ Mode continu : le bouton  lance l'exécution en mode continu
(comme s'il était lancé dans une boucle infinie)
- ✓ le bouton  arrête l'exécution (abandonner l'exécution)
- ✓ Le bouton  Animer l'exécution permet de visualiser le déroulement du programme dans le diagramme
- ✓ Sécuriser votre code VI

Palettes : Palette d'outils

- Palette d'outils est un élément commun entre les 2 fenêtres,
- Elle permet de manipuler, relier, placer les objets.



LED = allumée veut dire sélection automatique des Objets. Sinon il faut utiliser la touche tabulation du clavier



permet de **modifier la valeurs des objets**



permet de **sélectionner des objets** pour les déplacer, les modifier ou les supprimer



permet de **modifier le texte associé à un objet**



permet de **relier des objets entre eux** par des fils (dans le *diagramme*)



Outil sonde permettant d'afficher les valeurs qui transitent entre deux objets connectés



Outil coloriage

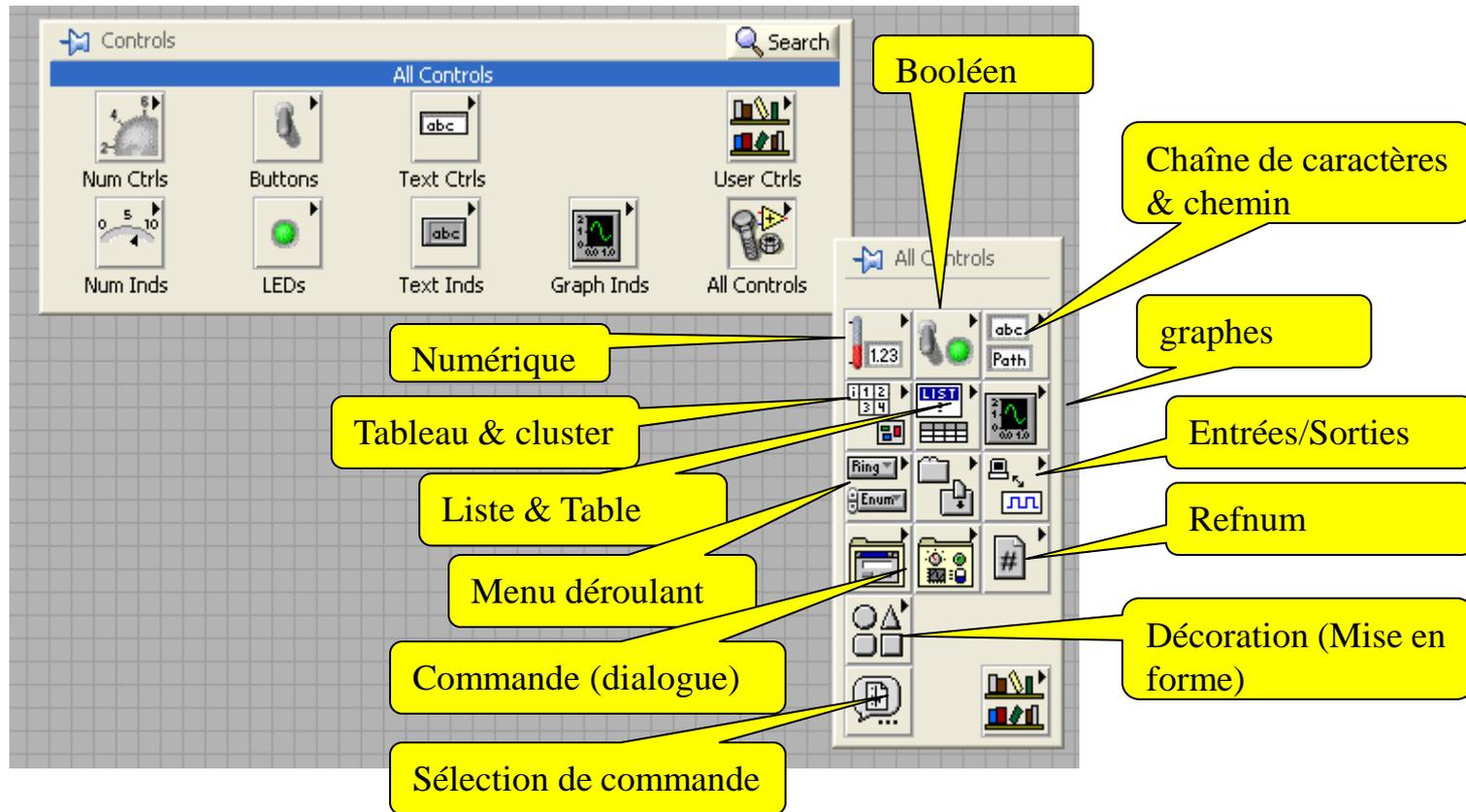


Mode Pause



Palettes : Palette de commandes

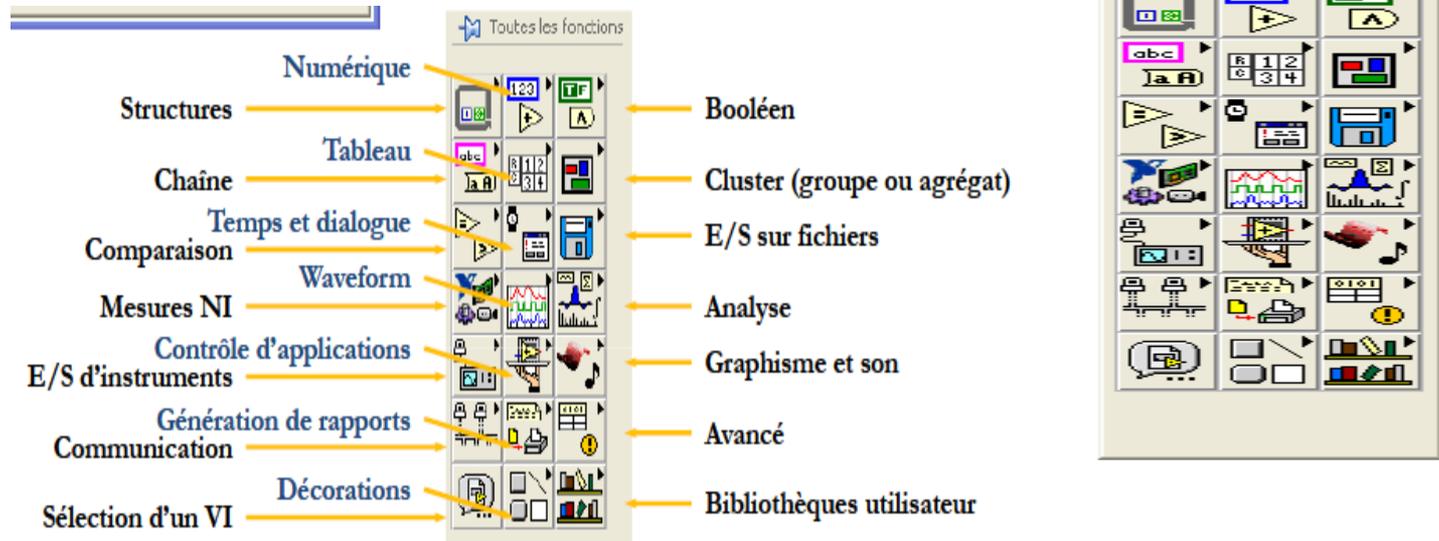
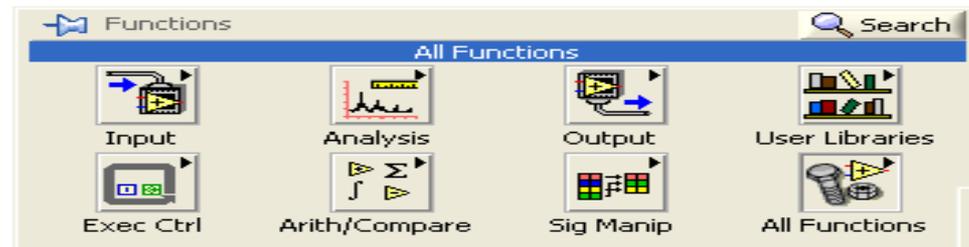
- Elle concerne la fenêtre face avant (clique droit)
- les boutons poussoirs, les zone d'affichage, potentiomètres...
- deux type d'objets : commande, indicateur.



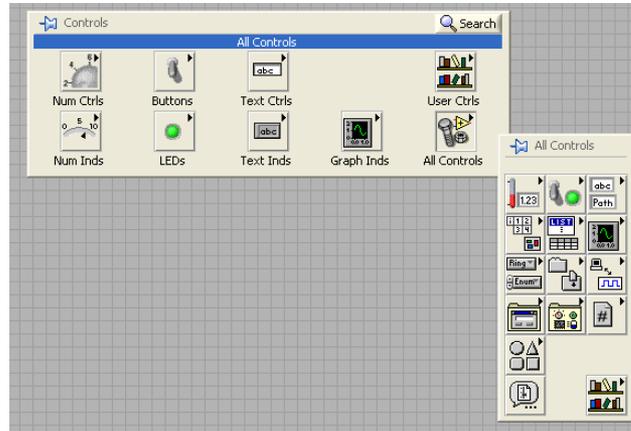
LABVIEW

Palettes : Palette de fonctions

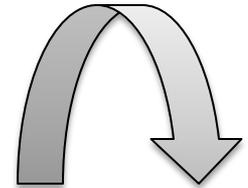
- Elle concerne la fenêtre diagramme (clique droit)
- les fonctions mathématiques, fonctions booléennes, structures, ...
- Ces fonctions constituent un programme



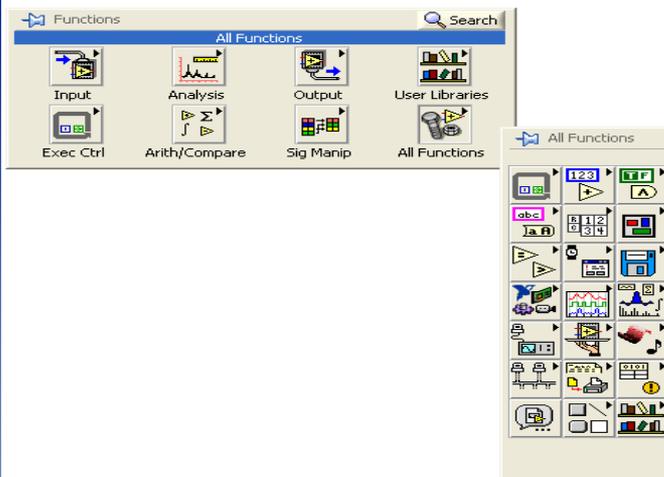
Création d'un VI (Virtual Instrument)



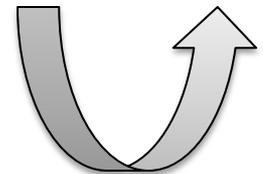
(Palette commandes)



Programme VI

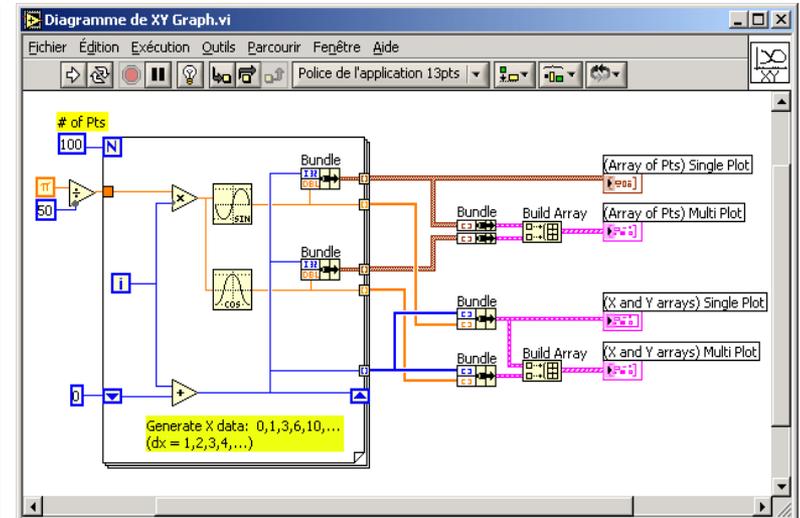
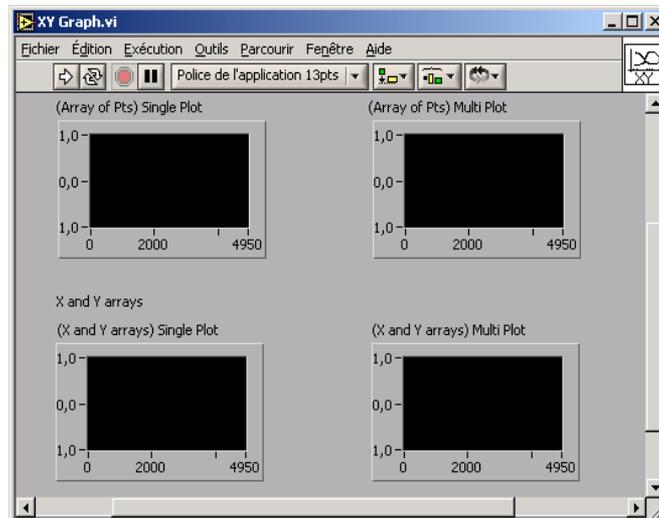


(Palette fonctions)



Méthode de Création d'un VI

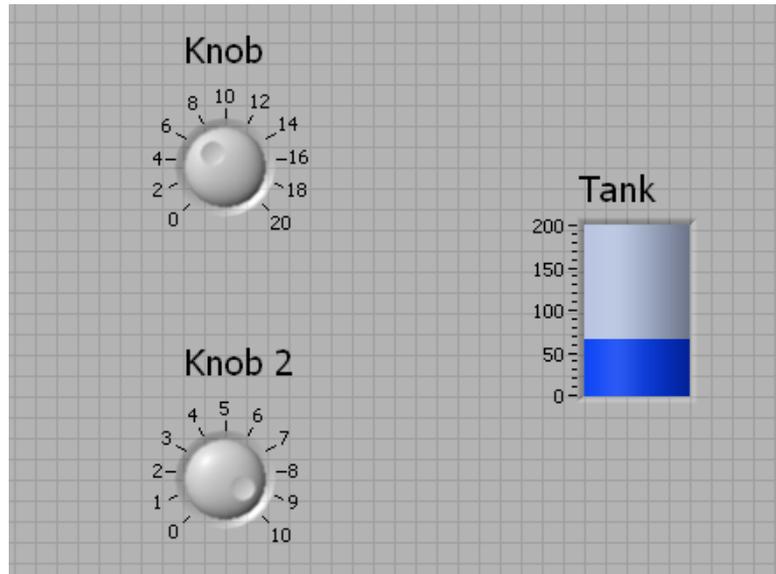
- Deux étapes distinctes :
 - ✓ On place les objets nécessaires sur la face avant.
 - ✓ On passe ensuite sur le diagramme, et on ajoute d'autres objets en les sélectionnant dans la palette fonctions. Ensuite, on relie les différents objets grâce à l'outil Bobine.



Exemple de Création d'un VI

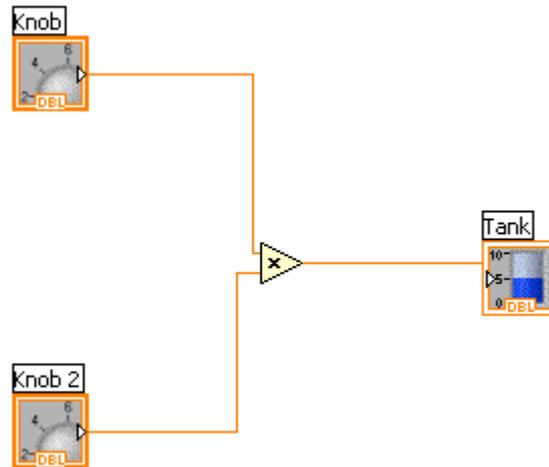
Afficher la multiplication de deux variables

Étape 1 : création des objets dans la face avant



Exemple de Création d'un VI

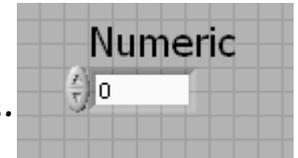
Étape 2 : créer le code dans la face diagramme



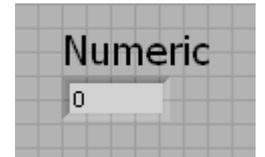
Principaux objets : palette commandes

➤ Objets numériques : deux types

✓ Numérique/commande numérique : saisie, ...



✓ Indicateur /indicateur numérique : affichage



□ Représentation : la taille mémoire

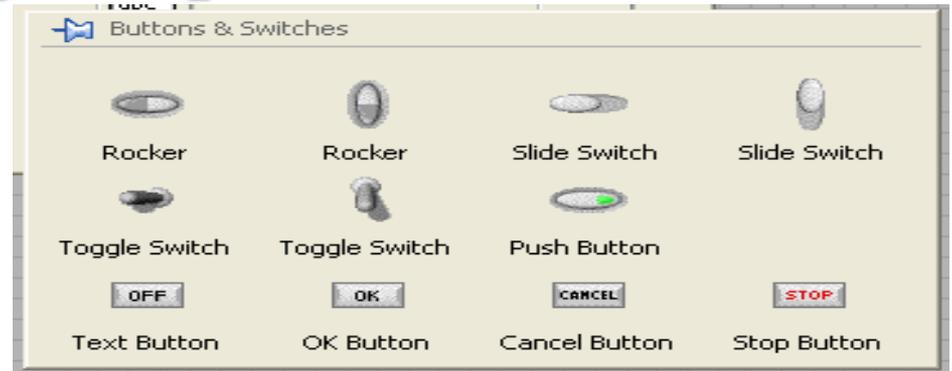
- SGL (Single précision), DBL (Double précision), EXT (Extended précision) : nombre réel (**Représentation en couleur Orange**)
- I8, I16, I32 : nombre entier relatif (**Représentation en couleur Bleue**)
- U8, U16, U32 : nombre entier naturel (**Représentation en couleur Bleue**).

Principaux objets : palette commandes

Type de fil de liaison	Scalaire	Tableau 1D	Tableau 2D	Couleur
Numérique	 	 	 	Orange (virgule flottante), Bleu (entier)
Booléen				Vert
Chaîne				Rose

Principaux objets : palette commandes

➤ Boutons poussoirs



✓ Propriétés mécaniques des boutons

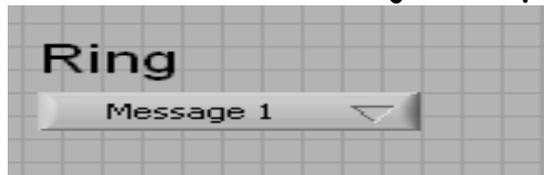
- Commutation à l'appui : le bouton change d'état si on clique dessus
- Commutation au relâchement : le bouton change d'état si on clique puis on relâche la souris
- Commutation jusqu'au relâchement : le bouton change d'état si on clique dessus, puis il revient à son état initial dès qu'on relâche la souris
- Armement à l'appui : le bouton change d'état si on clique dessus, puis revient à son état initial (même si la souris n'est pas relâchée)
- Armement au relâchement : le bouton change d'état si on clique puis on relâche la souris, ensuite il revient tout de suite à son état initial
- Armement jusqu'au relâchement : le bouton change d'état si on clique dessus, puis il revient à son état initial peu de temps après qu'on a relâché la souris

Principaux objets : palette commandes

➤ Chaînes de caractères : deux types

- ✓ string indicator : pour afficher un message
- ✓ string control : pour saisir un message texte
- ✓ On peut également utiliser des menus déroulants (Menu Ring)

Menu déroulant représente une paire de valeurs (une chaîne et un numérique) qui peut faire partie d'une liste de valeurs. L'enchaînement des valeurs peuvent être séquentiel ou pas contrairement à enums où l'enchaînement est séquentiel et il commence toujours par la valeur 0.



Exemple : Utiliser le menu déroulant (Ring) et afficher le message
- ENSA de Tanger ou Filière GIND1 ou Labview

Principaux objets : palette commandes

➤ Tableaux et clusters

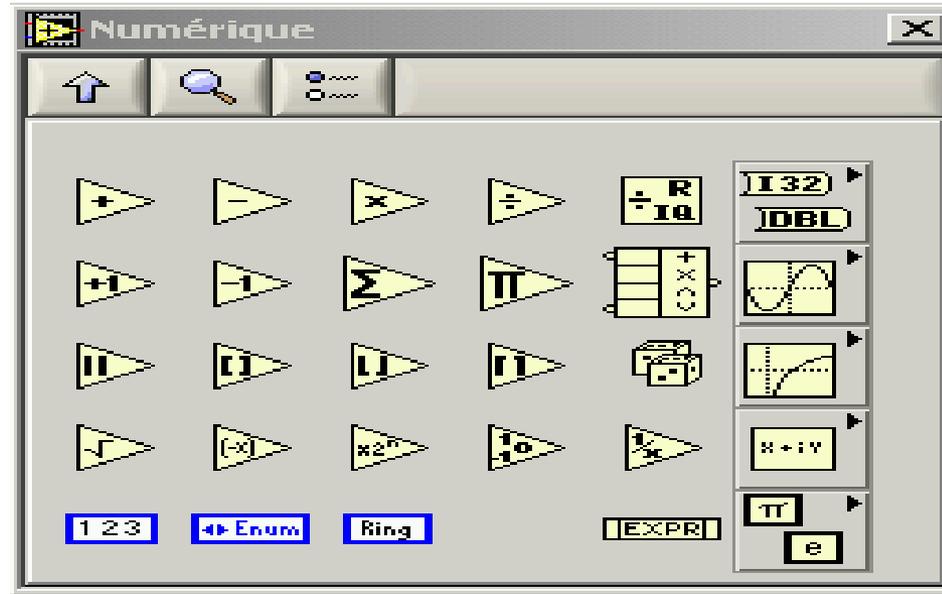
- ✓ Tableaux groupent des éléments du même type
- ✓ Clusters groupent des données de types différents

Exemple 1: Reprendre l'exemple précédent en utilisant un tableau de dimension 1.

Exemple 2: Reprendre l'exemple précédent en utilisant l'objet Combo Box de la palette String & Path

Principaux objets : palette fonctions

➤ Fonctions et constantes arithmétiques

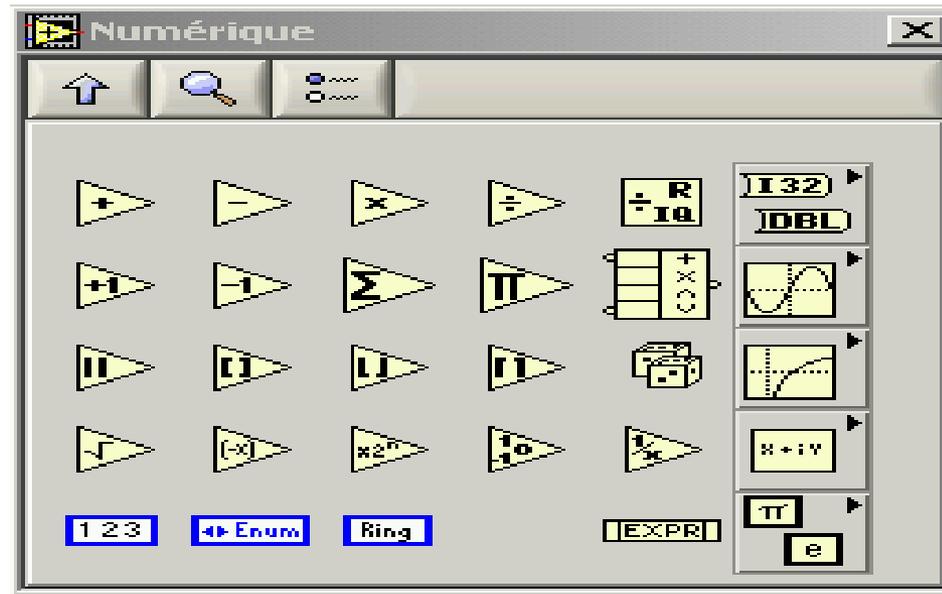


Exemple 1 :

afficher la somme et le produit avec deux méthodes, la valeur entière la plus proche, la valeur arrondie inférieure, la valeur arrondie supérieure du produit de :
(1, 7, 8, 9, 10, 5.5 et un objet numérique de type contrôle)

Principaux objets : palette fonctions

➤ Fonctions et constantes arithmétiques



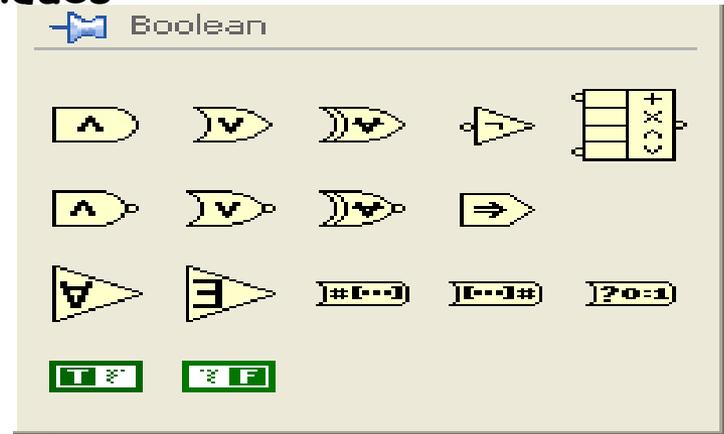
Exemple 2 :

Générer une variable aléatoire entre 0 et 1.

LABVIEW

Principaux objets : palette fonctions

➤ Fonctions et constantes logiques



Exemple 1 :

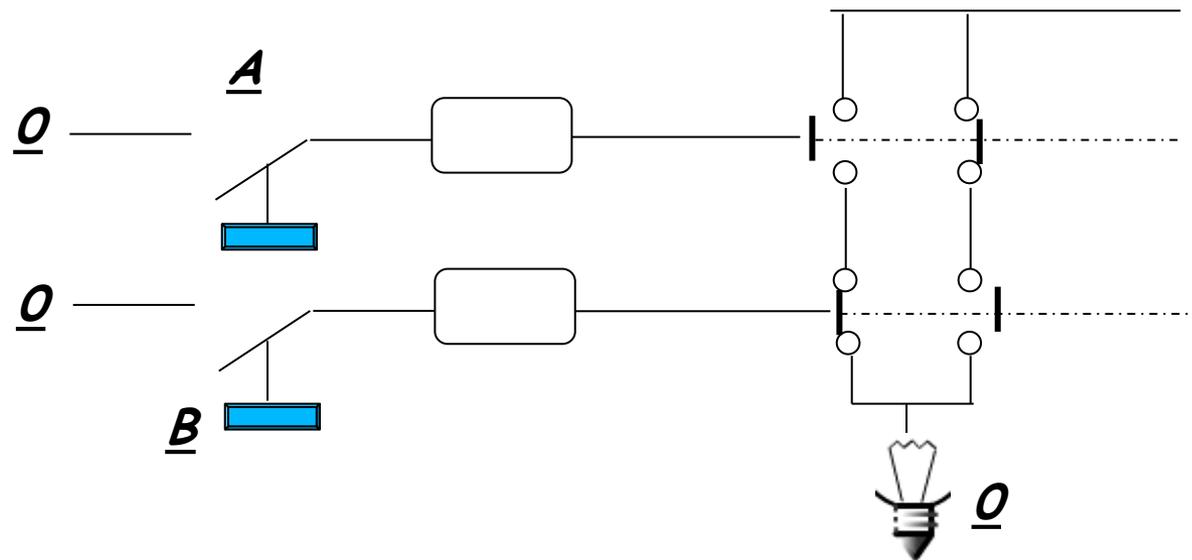
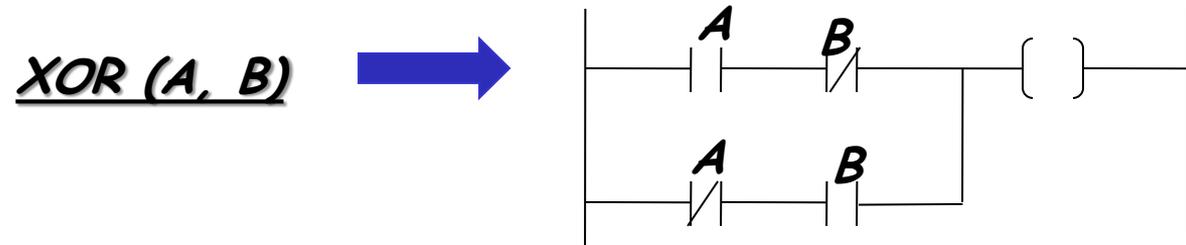
Convertir l'état True ou False d'un bouton poussoir en 1 ou 0.

Exemple 2 : Tableau de vérité

p	Vrai	Vrai	Faux	Faux
q	Vrai	Faux	Vrai	Faux
$p \Rightarrow q$	Vrai	Faux	Vrai	Vrai

Utiliser la fonction implique et montrer son équivalence avec : $Non(P)$ ou Q .

Exemple : XOR

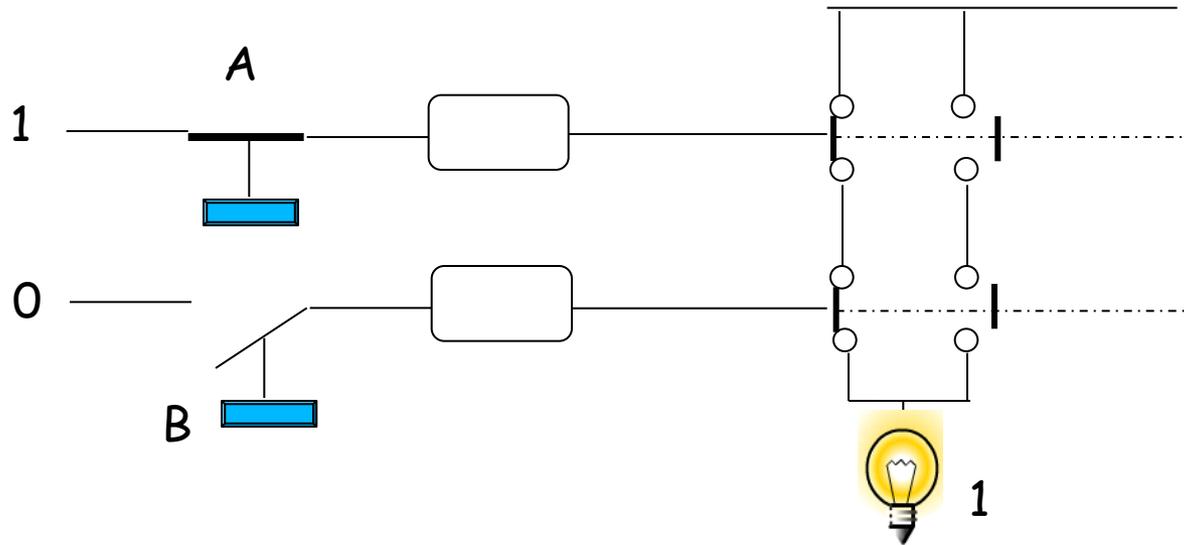
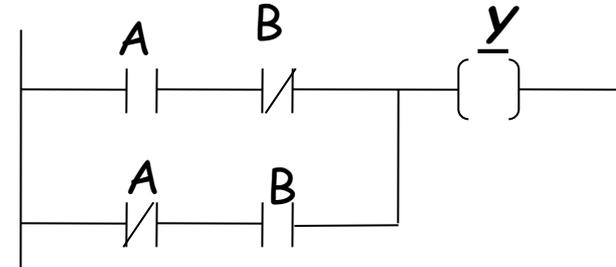


Exemple : utiliser la fonction XOR avec deux booléens de type contrôle et Un objet LED comme indicateur

LABVIEW

Exemple : XOR

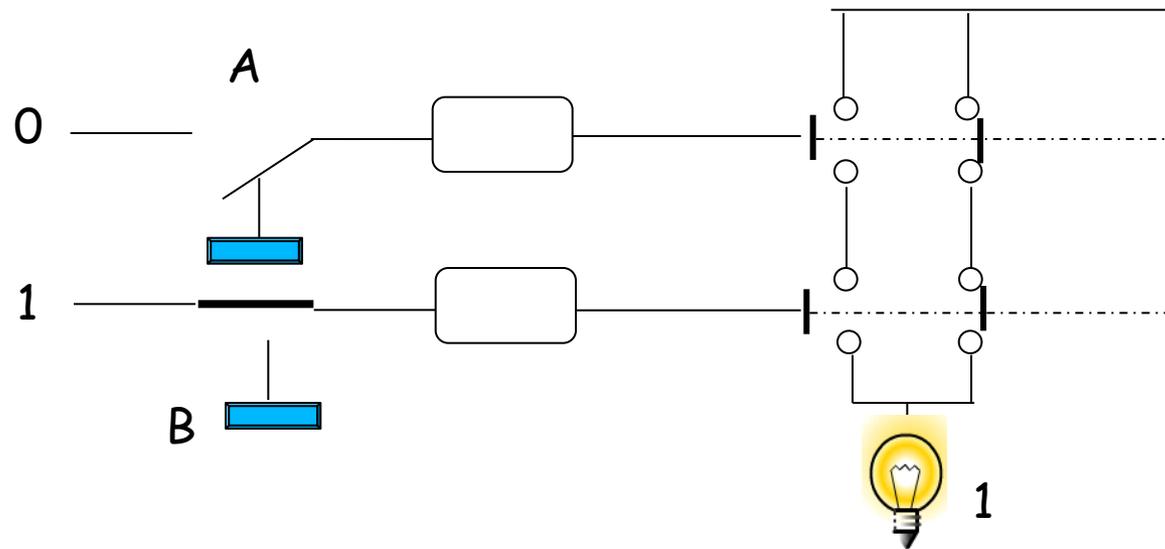
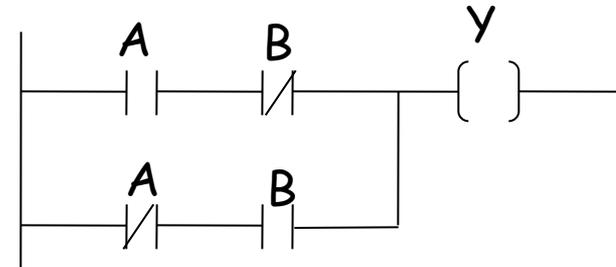
XOR (A, B) →



LABVIEW

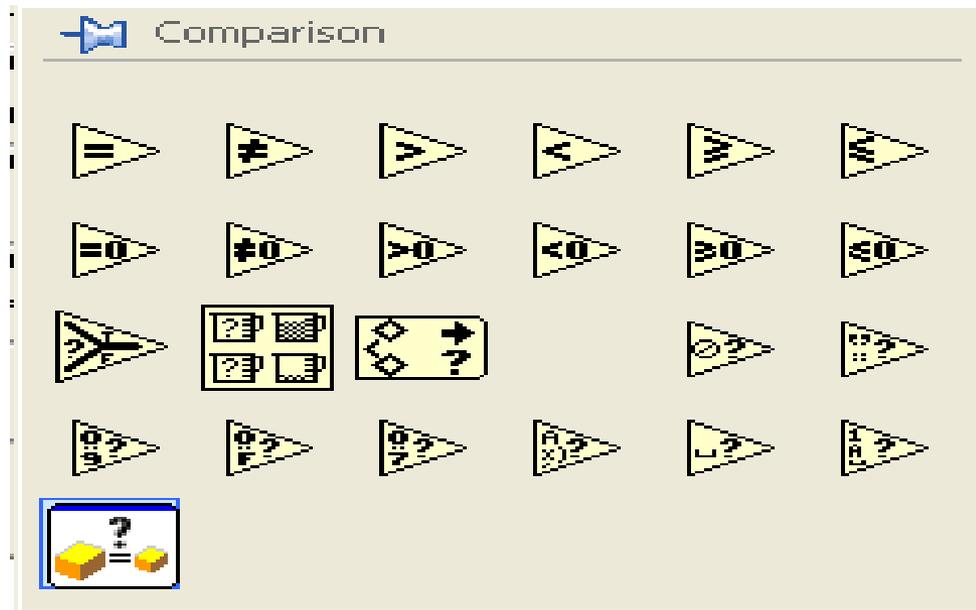
Exemple : XOR

XOR (A, B) →



Principaux objets : palette fonctions

➤ Fonctions de comparaison



Exemple :

- Afficher le max et le min de deux nombres
- En utilisant la fonction *select* créer un VI permettant d'afficher l'état d'un bouton poussoir (True, False).

Principaux objets : palette fonctions

➤ Fonctions attendre



(Unité de temps : ms)

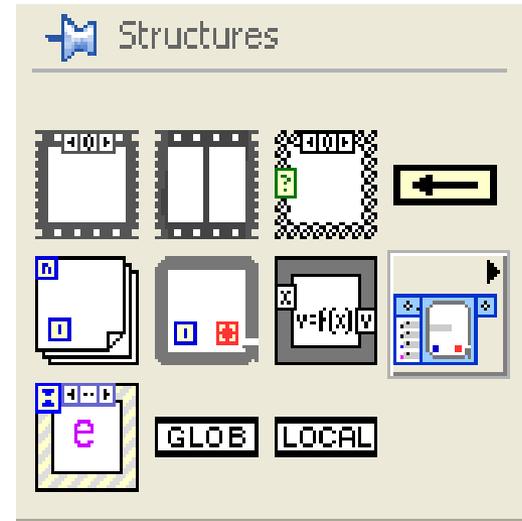


- Cette fonction permet, pendant l'exécution, de créer une temporisation pendant un certain temps spécifié en millisecondes.
- La valeur d'entrée contrôle cette temporisation.



(Unité de temps : s)

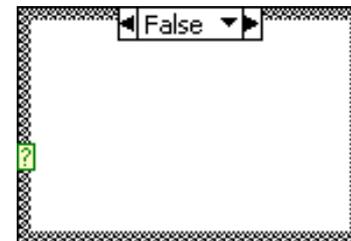
➤ Structures de contrôle



✓ structure condition



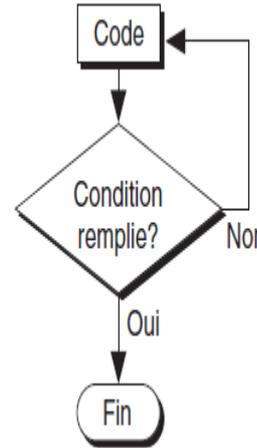
Si-Alors-Sinon



Exemple : Afficher le message «le niveau est atteint » **Si**
 Le niveau du liquide d'un Tank (type contrôle) est = 10
Sinon afficher dans la même zone le message
 « pas encore »

Structures de contrôle

✓ Structure While



Répéter (code);

Jusqu'à ce que la condition soit remplie;

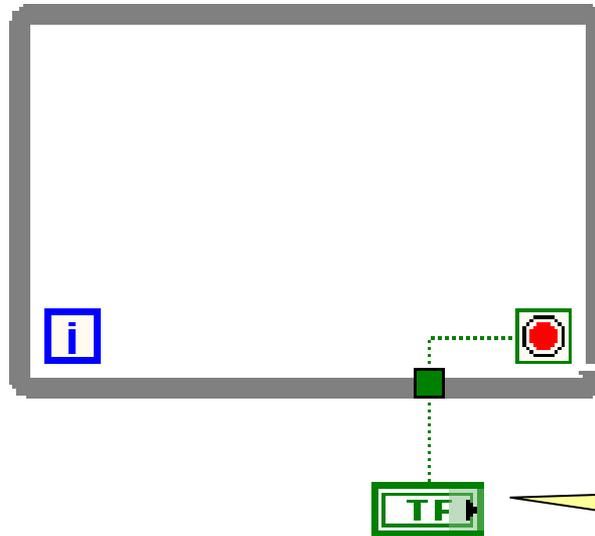
Fin;

- ✓ le bloc d'instructions placé dans le carrée est répété tant qu'une condition est vérifiée.
- ✓ la condition doit être reliée au symbole 
- ✓ le compteur de la boucle est le symbole 
- ✓ une boucle while infinie est une boucle dont la condition est toujours vraie.
- ✓ La boucle While s'exécute toujours au moins une fois.

Exemple : la consigne de la température d'un four pour chauffer un lot de pièces est de 80°. Créer un VI permettant d'incrémenter par 1 la valeur d'un capteur thermomètre tant que la consigne n'est pas atteinte.

Structures de contrôle

- ✓ Structure While



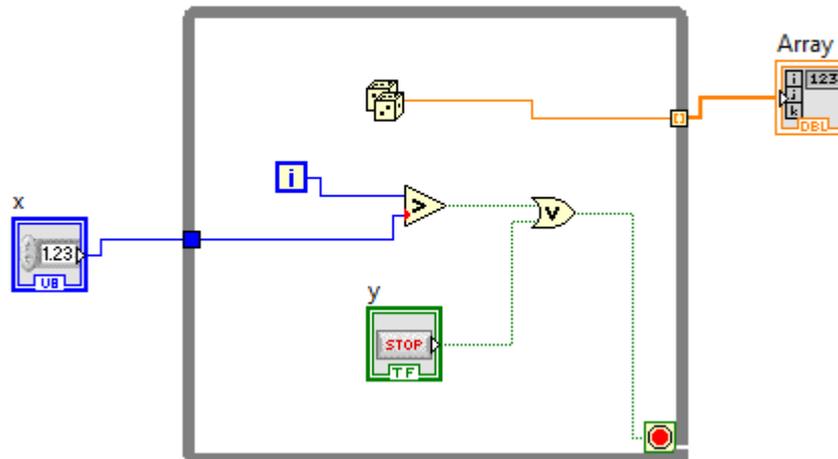
Ce terminal est initialement à faux

Combien de fois s'exécute cette boucle ?

Structures de contrôle

✓ Structure While et les Tunnels

- Les tunnels acheminent les données vers l'intérieur et l'extérieur .
- Lorsqu'un tunnel transmet des données à une boucle, celle-ci ne s'exécute qu'une fois les données arrivées au tunnel.
- Les données sortent de la boucle une fois l'exécution de la boucle achevée

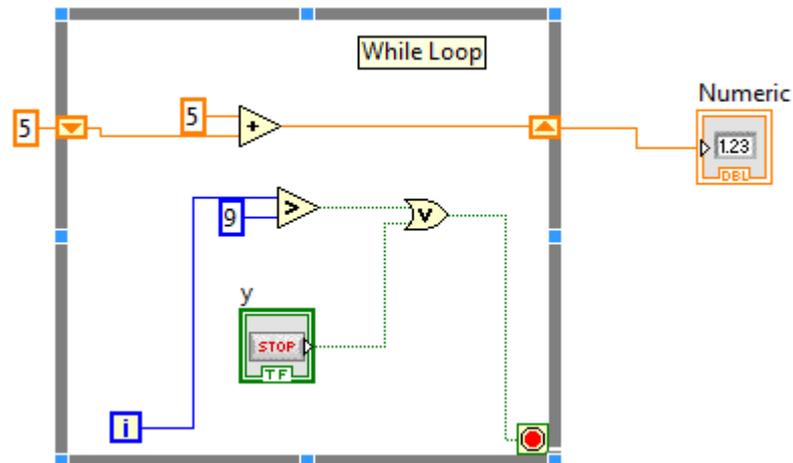


Structures de contrôle

✓ Structure While et Registres à décalage

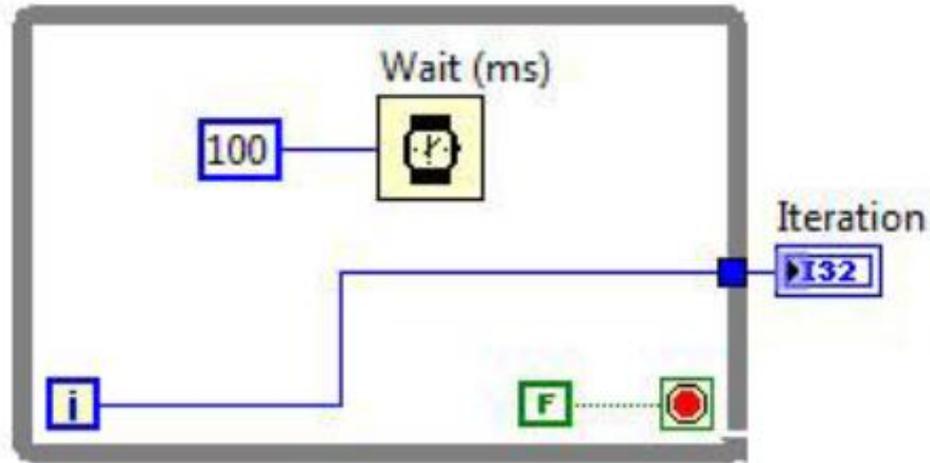
Exemple

- Le registre à décalage sont caractérisés par deux flèches : entrante et sortante
- Le registre à décalage est initialisé à la valeur 5.
- la flèche d'entrée reçoit les valeurs de la flèche de sortie
- à la fin de l'exécution de la boucle, l'objet numérique reçoit la valeur somme.



LABVIEW

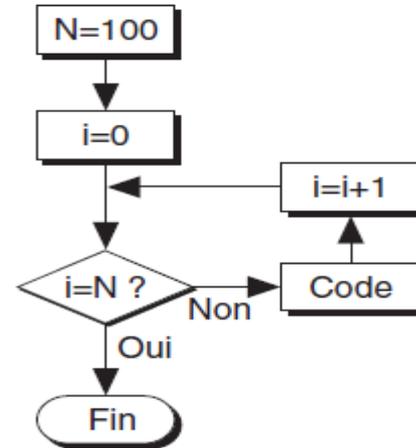
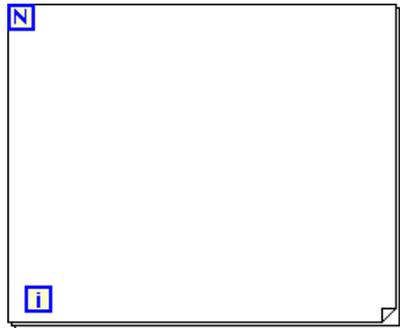
Which of the following statements is TRUE about the following code segment?



- a. The loop will execute once and the indicator Iteration will have a value of one*
- b. The loop will execute once and the indicator Iteration will have a value of zero*
- c. The loop will execute infinitely and the program will have to be aborted*
- d. The loop will not execute and the indicator Iteration will have a value of zero*

➤ Structures de contrôle

- ✓ Structure For : le même principe que la boucle While



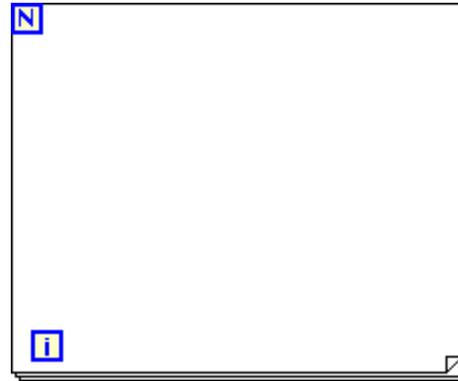
- ✓ On doit spécifier la valeur maximale jusqu'à laquelle doit aller le compteur
- ✓ Cette valeur est associée au symbole

Exemple 2 : Créer un VI permettant d'afficher d'une part dans un tableau 1D les 20 valeurs générées par un Dé et d'autre part dans un autre objet numérique la dernière valeur générée.

Exemple 3 : En utilisant la boucle For, créer maintenant un autre VI qui permet d'afficher dans un tableau 2D les 30 valeurs générés par un Dé.

➤ Structures de contrôle

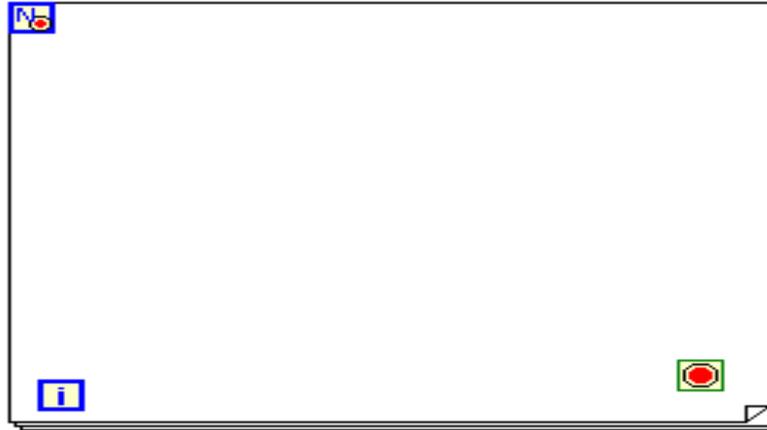
- ✓ Structure For : le même principe que la boucle While



- ✓ On doit spécifier la valeur maximale jusqu'à laquelle doit aller le compteur
- ✓ Cette valeur est associée au symbole 

Exemple : on fait exécuter une boucle For 100 fois et a chaque fois on calcule le cosinus de la valeur du compteur (en radians) que l'on branche sur un objet « Sinusoïde » de type Graphe déroulant.

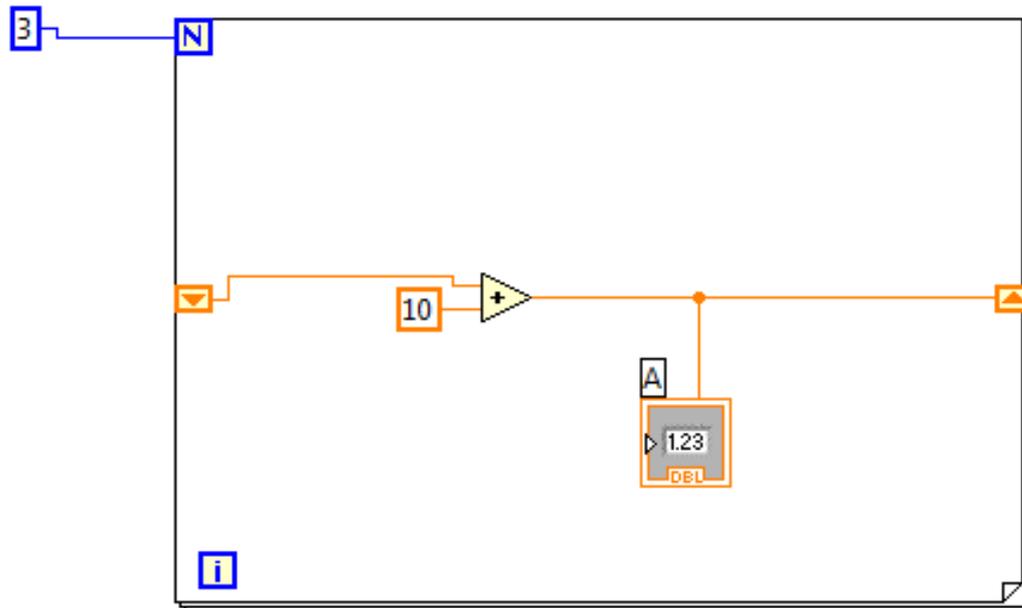
- Structures de contrôle For avec l'ajout d'une condition
 - ✓ Structure For : le même principe que la boucle While



- ✓ On peut ajouter une condition à une boucle for.
- ✓ Pour ajouter un terminal de condition à une boucle For, cliquez avec le bouton droit sur le bord de la boucle For et sélectionnez Terminal de condition dans le menu local. Câblez ensuite le terminal de condition et le terminal de décompte. Dans ce cas si l'une des conditions est vérifiée l'exécution s'arrête

➤ Registres à décalage et Nœud de retour

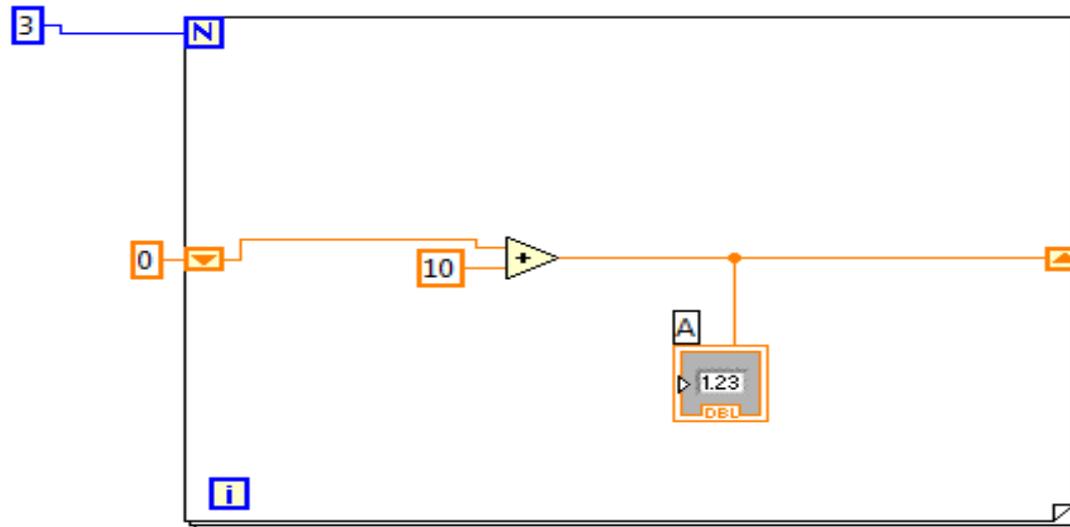
Exemple



➤ Registres à décalage

Exemple

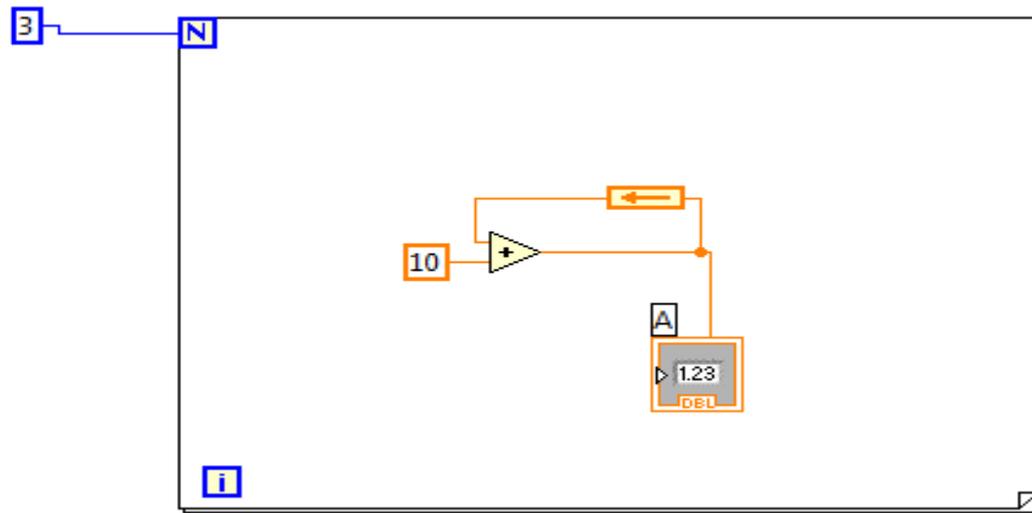
Initialiser la somme à chaque exécution en utilisant la boucle For et la registre à décalage



➤ Nœud de retour

Exemple

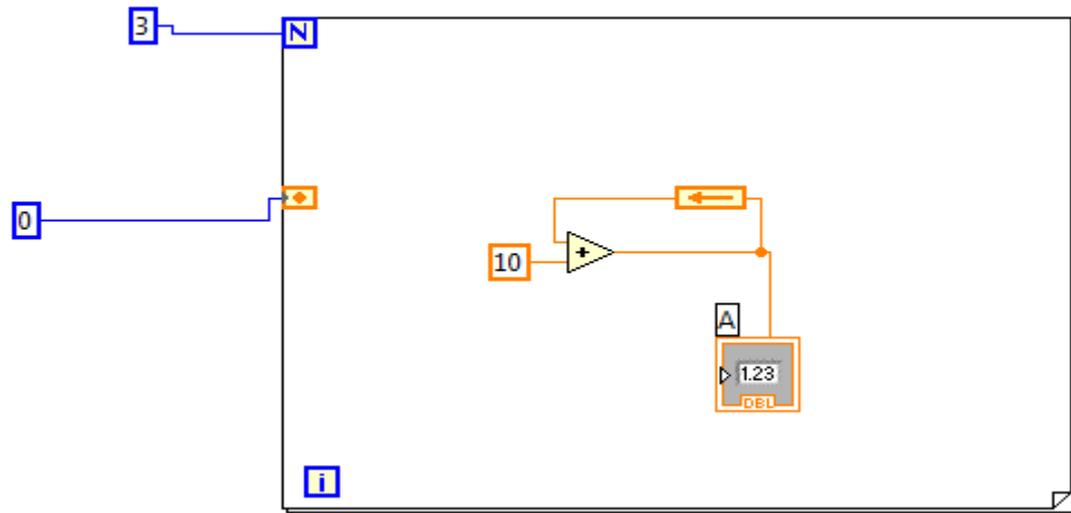
Créer un objet numérique de type indicateur,
 Dans la fenêtre diagramme, placer la fonction nœud de retour de la
 Palette structures.



➤ Nœud de retour

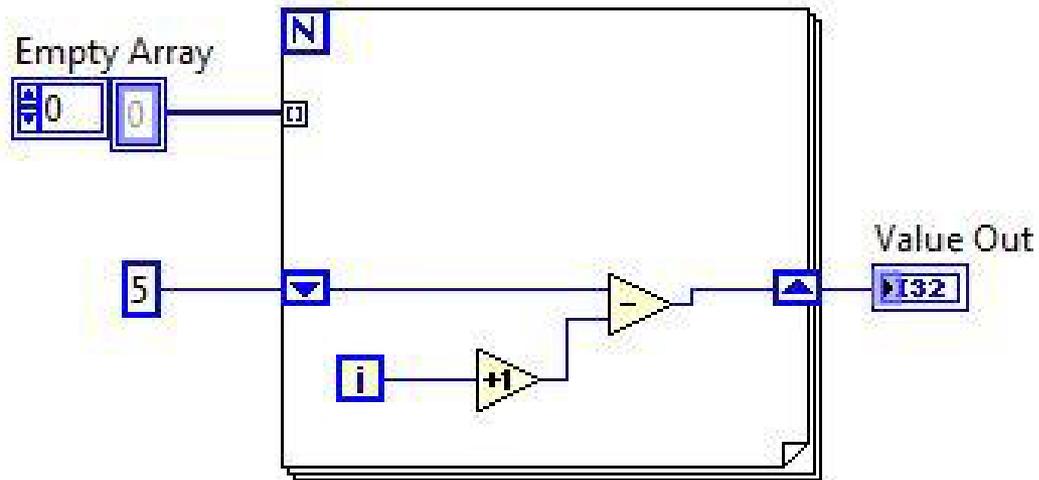
Exemple

Initialiser la somme pour chaque exécution du programme
 Clic droit sur la fonction nœud de retour et choisir initialiser le Terminal



➤ Structures de contrôle For

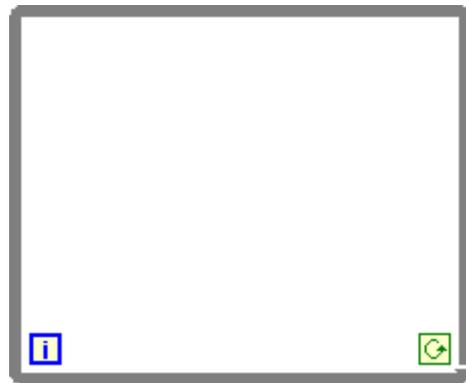
What value does the Value Out indicator display after the VI executes?



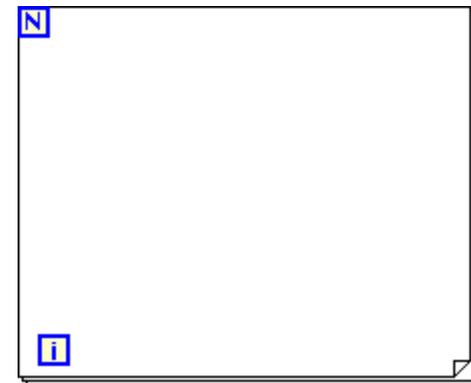
- a. 0
- b. 4
- c. 5
- d. 6

➤ Structures While and For

✓ Structure While



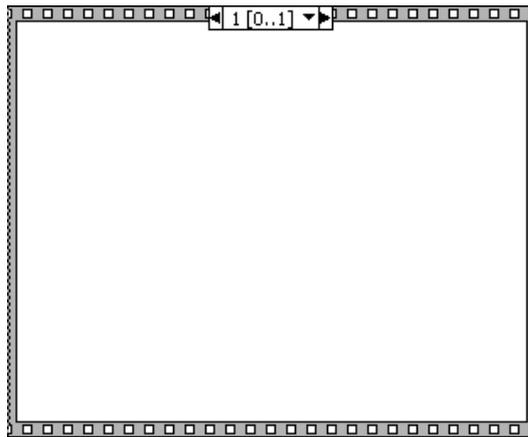
✓ Structure For



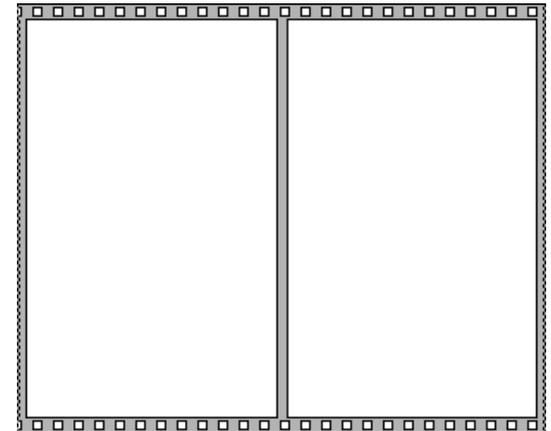
Exemple 1 : En utilisant les deux boucles While et For, créer un VI pour afficher :

- 10 valeurs aléatoires entre 0 et 1.
- les valeurs max, min et moyenne du tableau (utiliser la palette Signal Analysis/Statistics).
- les valeurs qui sont supérieures à 0.3 dans l'ordre croissant.

- Structures de contrôle
 - ✓ Structure séquences



Ou

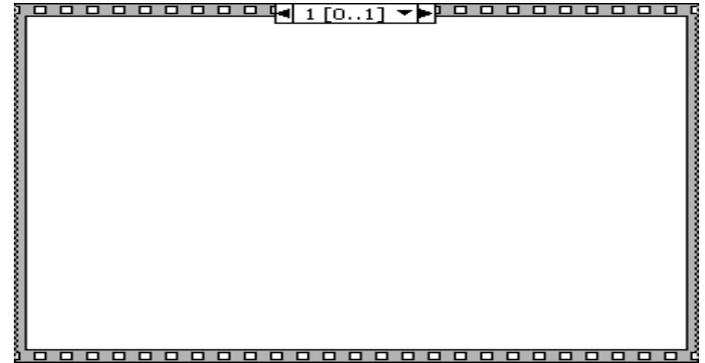


✓ La structure Séquence est utilisée lorsque on souhaite exécuter des actions de manière séquentielle

Exemple : Créer une *séquence* comportant 3 étapes numérotées de 0 à 2, permettant d'allumer deux boutons de type LED à 1 seconde d'intervalle

➤ Structures de contrôle

- ✓ Structure séquences

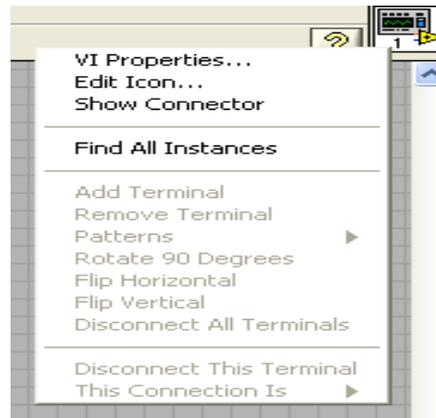


- ✓ La structure Séquence est utilisée lorsque on souhaite exécuter des actions de manière séquentielle

Exemple 2 : En utilisant la structure de contrôle séquence, créer un VI qui permet de gérer les sémaphores d'un carrefour

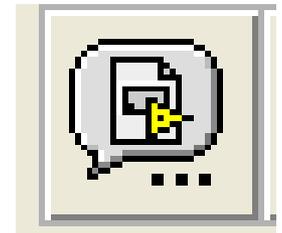
➤ Création et utilisation de « sous-VI »

- ✓ Tout VI peut devenir un sous VI ou un sous programme
- ✓ étapes de création d'un sous VI.
 - étape 1 facultative : Editer l'icône de la fenêtre face avant par clic-droit (modifier le dessin créer par défaut)
 - étape 2 obligatoire show Icon (visualiser le connecteur)



➤ Appeler le sous VI

- ✓ utiliser la palette Fonctions / Select a VI... (fenêtre diagramme)



➤ **Variable locale, registres à décalage et le nœud de retour**

✓ Lorsqu'on veut utiliser la même variable plusieurs fois

Exemple 1 :

Créer un VI pour afficher dans la variable A la somme de A +10.

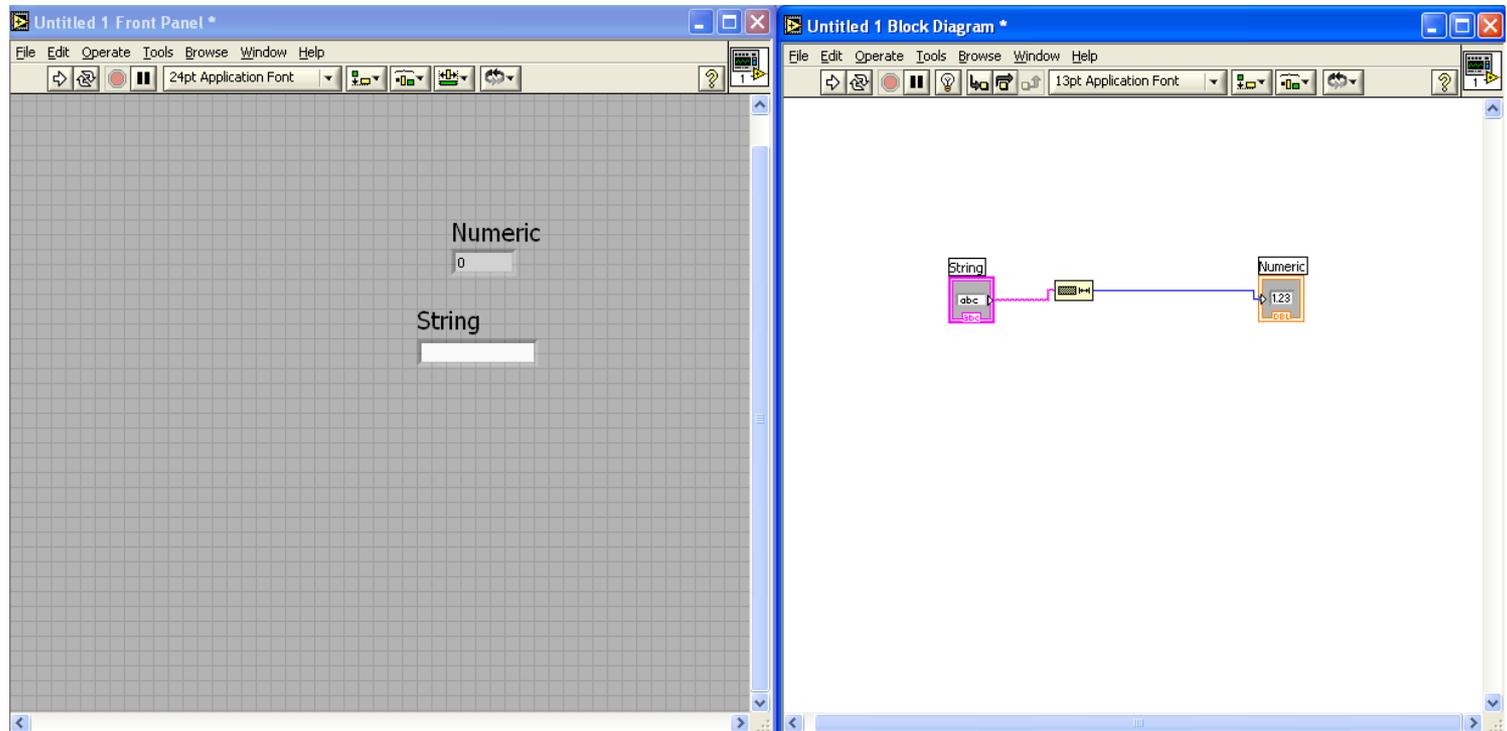
Trois solution possibles :

- Variables locales
- registres à décalage (boucles While et for)
- nœud de retour (boucles While et For)

Exemple 2 : reprendre l'exemple du slide 25 pour afficher le troisième message (le niveau est strictement supérieur à 10 **Si** le niveau du liquide dans le tank est > 10) dans la même zone si le niveau dans le tank dépasse la valeur 10.

➤ chaîne de caractère

- ✓ longueur d'une chaîne
 - un objet numérique de type indicateur :
affichage la longueur
 - un objet chaîne de caractère de type contrôle
 - la fonction longueur



➤ chaîne de caractère

✓ Concaténer deux chaînes

Exemple : concaténer les deux chaînes suivantes :

- la filière
- GIND1

✓ extraire des caractères d'une chaîne

- la fonction sous ensemble permet d'extraire un ou plusieurs caractères. On doit spécifier en entrée
 - la chaîne de départ
 - la position à partir de laquelle on va extraire (offset)
 - le nombre de caractères à extraire (longueur)

Exemple :

Automatique VI  *matique*

- R/W sur les registres
- Advanced/Accès au contenu de la mémoire
 - ✓ Ecrire dans un registre :



Deux entrées :

l'adresse du registre dans lequel on veut écrire (register address)
et la donnée à écrire (value).

- ✓ Lire dans un registre :



Il prend en entrée l'adresse du registre
et renvoie la valeur stockée