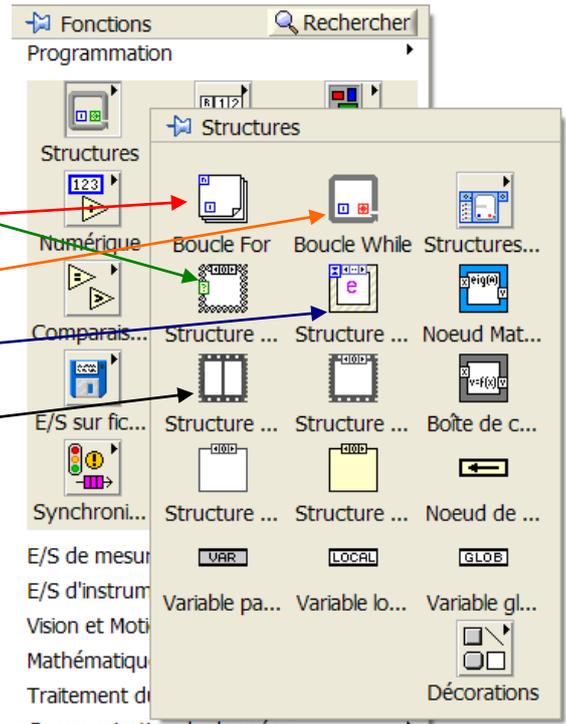


1) Structures :

Il existe 5 principaux types de structures sous labview :

- ✚ La condition : Si alors sinon
- ✚ La boucle For : Faire ... pour
- ✚ La boucle While : Faire ... tant que
- ✚ La structure évènement
- ✚ La structure de blocs séquence.



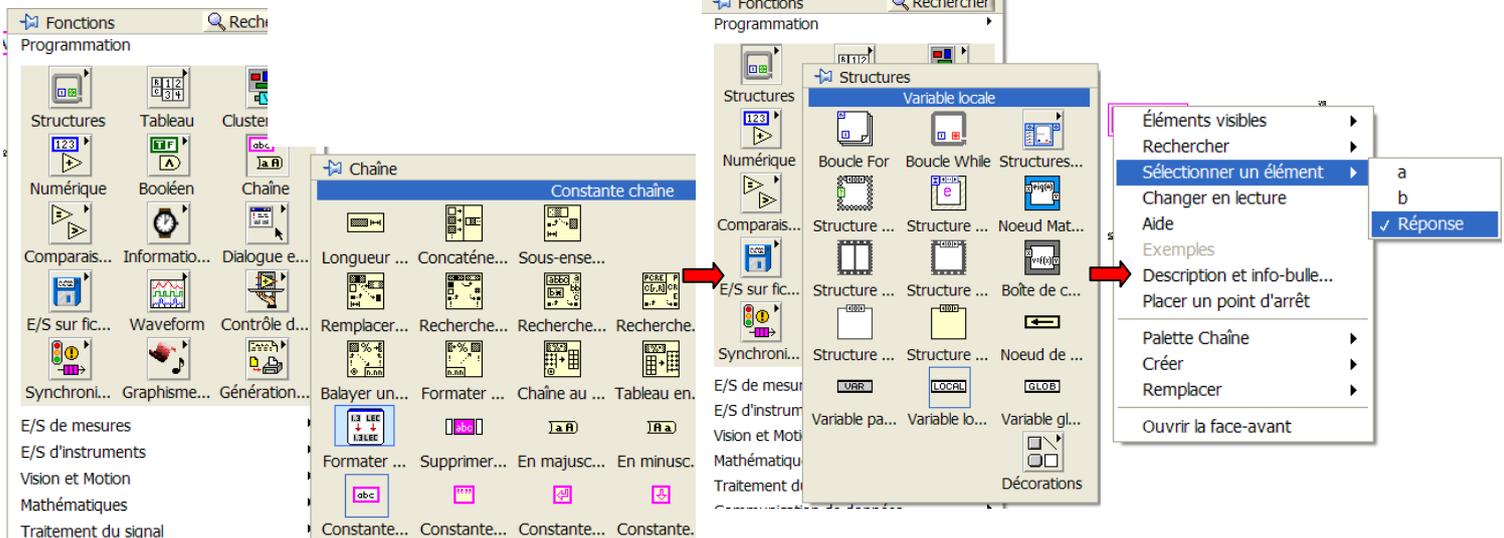
Les structures de programmation se trouvent dans la palette de fonctions

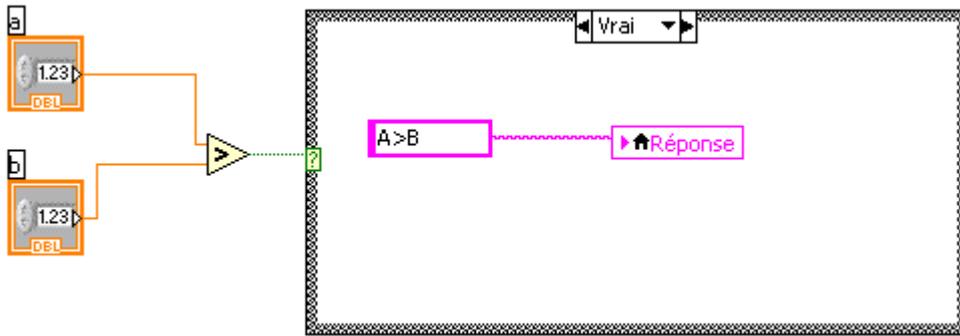
a) Structure condition :

Exemple de structure condition:

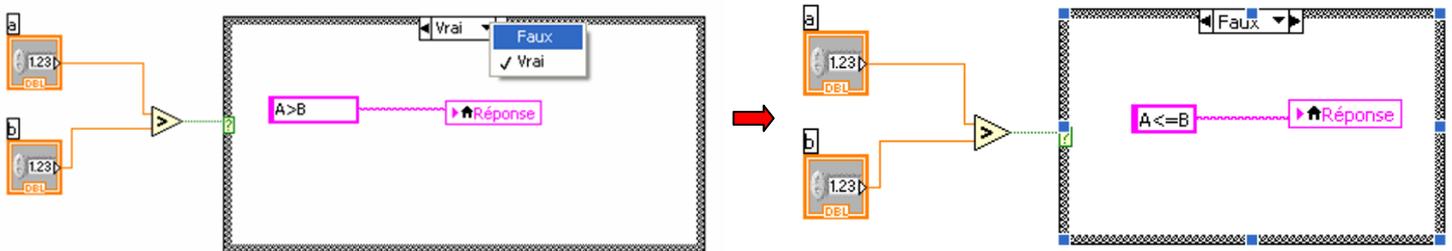
SI $A > B$ le programme réponds "A>B"
SINON (condition fausse) le programme réponds "A<=BB"

- ✚ On créé un nouveau VI
- ✚ Sur la face avant on ajoute deux commandes (a et b) et un indicateur de chaîne de caractères (réponse).
- ✚ Dans le diagramme :
 - On ajoute le comparateur « > »
 - on ajoute la structure condition.
 - on complète les deux conditions :
 - ✚ pour la condition vrai : on créé une constante chaîne (A>B) et une variable locale (Réponse : on clique sur la variable avec le bouton droit).

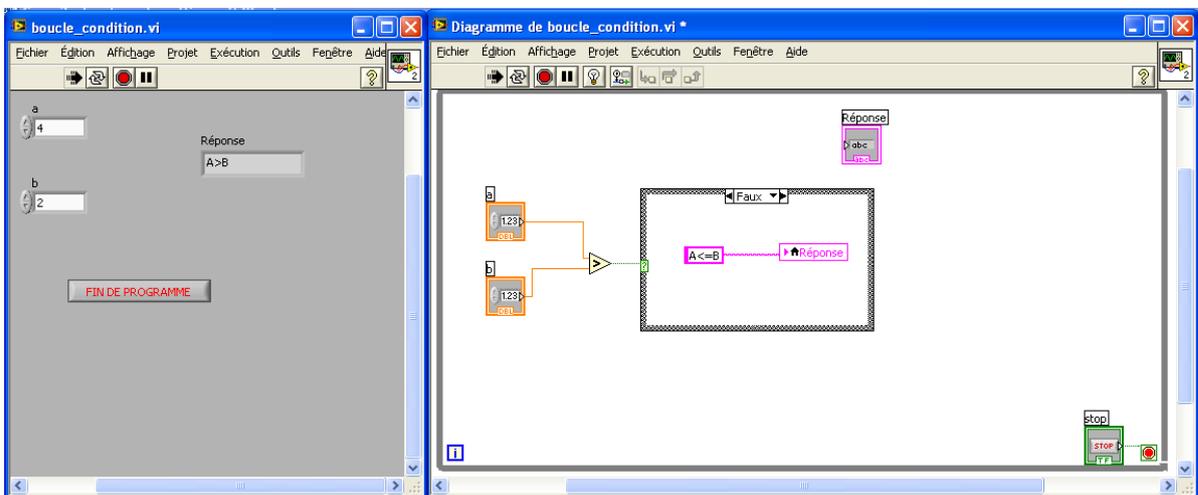




- ✚ pour la condition fausse (on clique avec le bouton droit sur l'indicateur à la fin du vrai et on choisit Faux): on crée une constante chaîne ($A \leq B$) et une variable locale (Réponse : on clique sur la variable avec le bouton droit).



- on câble sorties->entrées... Sorties -> entrées.
- on ajoute une structure de répétition (on crée le bouton stop en cliquant sur « créer une commande »).
- On obtient :



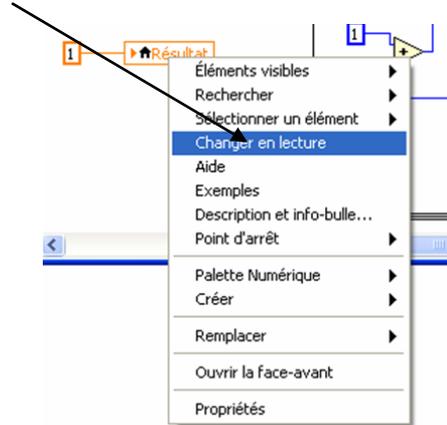
b) Boucle for (faire... pour) :

On répète une action un nombre de fois connu dès le début de la boucle.

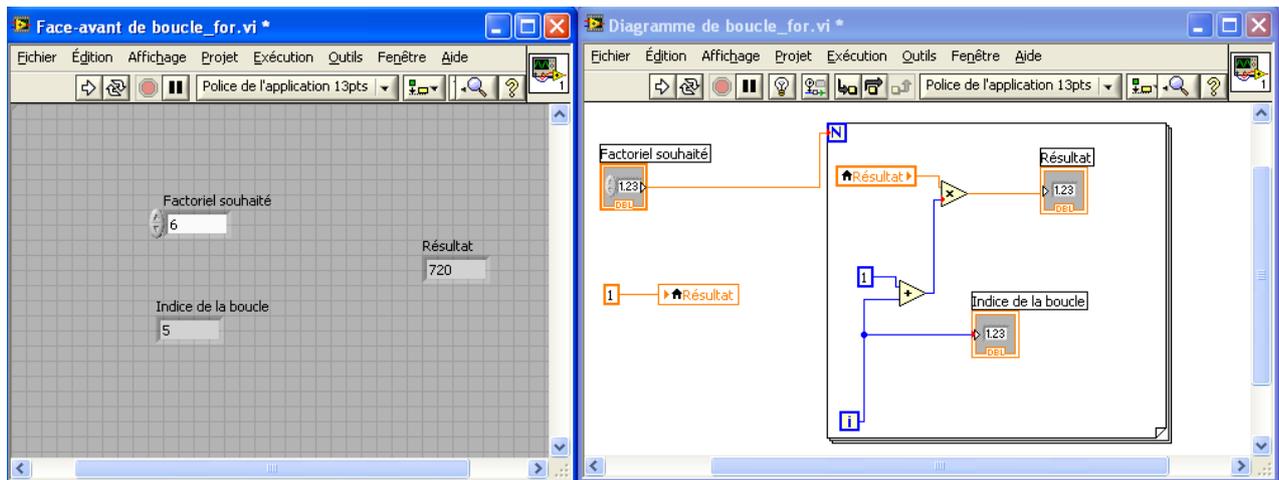
Exemple : calculons le factoriel d'un nombre

- ✚ On crée un nouveau VI
- ✚ Sur la face avant on ajoute une commande (factoriel) et deux indicateurs numériques (Indice de la boucle et Résultat)

- Dans le diagramme :
 - On ajoute la boucle for
 - On ajoute l'opérateur + et *
 - On ajoute deux fois un constante numérique = 1
 - On ajoute deux variables locales résultat (une en écriture à 1 pour initialiser le résultat et l'autre en lecture (le trait extérieur devient plus épais) pour incrémenter le factoriel)



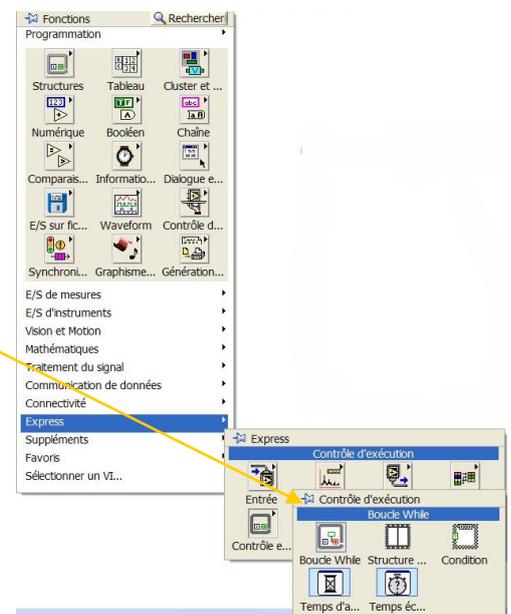
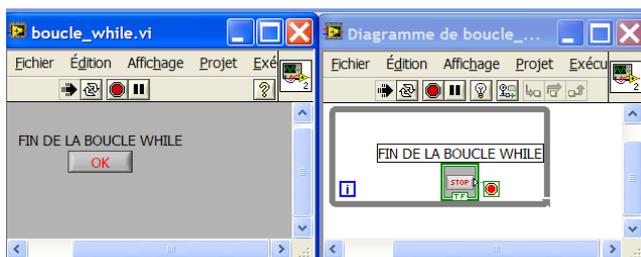
- on câble sorties->entrées... Sorties -> entrées.
- On obtient :



c) Boucle While (faire ... tant que)

C'est un rectangle à coin arrondi, la condition de fin est câblée sur le booléen de sortie (rond rouge). Les actions contenues dans la structure se déroulent tant que la condition de fin n'est pas vraie.

Dans les outils express, on trouve une boucle while avec le bouton stop déjà câblé.



d) Structure évènement :

La structure dite "évènement", est un bloc d'actions (similaire à la condition) effectuée lors de l'exécution d'une action telle que :

- ✚ un appui sur un bouton de face avant,
- ✚ un appui sur une touche,
- ✚ un temps écoulé,
- ✚ un appui sur un bouton de la souris.

Une fois la première structure créée, on ajoute le nombre de structures nécessaires (clic droit sur le bord de la structure) en choisissant l'évènement déclenchant dans la liste de la boîte de dialogue qui apparaît. On peut ensuite écrire les parties de programme correspondant à chaque bloc dans ceux ci.

Exemple :

On va créer un programme qui :

- ❖ ouvre une fenêtre avec écrit « bonjour » si on clique sur le bouton BONJOUR et affiche dans la chaîne de caractères BON CHOIX
- ❖ fait un bip si on clique sur le bouton BIP et affiche dans la chaîne de caractères BON CHOIX
- ❖ affiche dans la chaîne de caractères : MAUVAIS CHOIX si on sort avec la souris de la face avant.

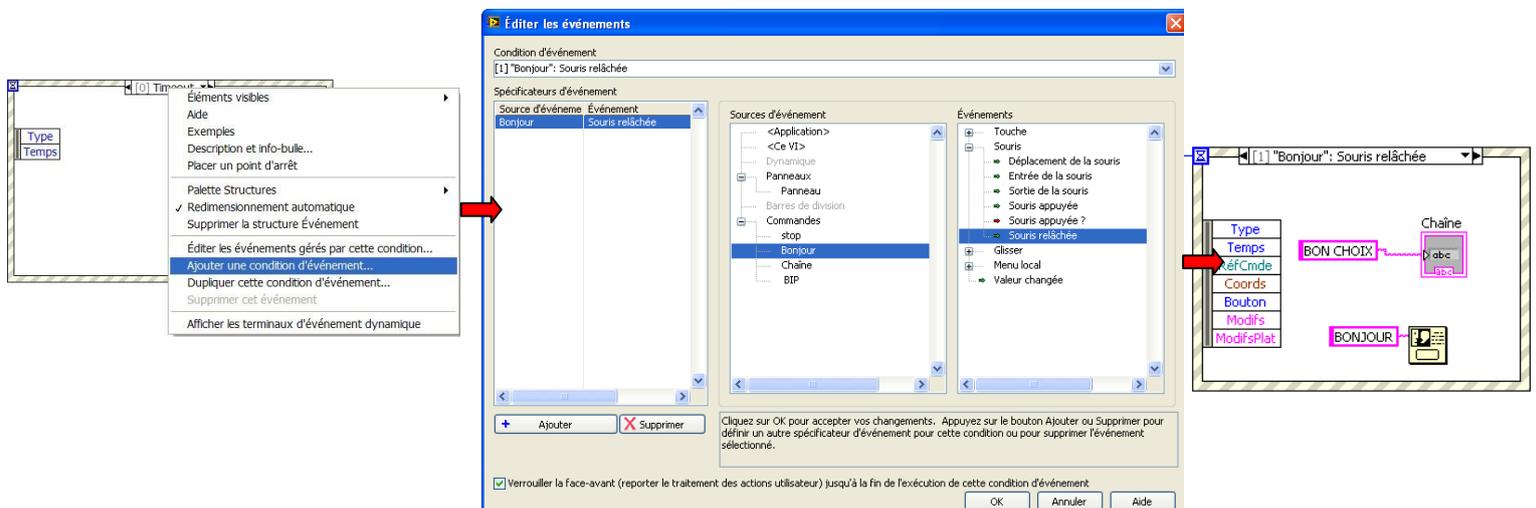
- ✚ On créé un nouveau VI
- ✚ Sur la face avant on ajoute trois boutons (BONJOUR, BIP et FIN DE PROGRAMME) et un indicateur de chaîne de caractères (CHOIX)
- ✚ Dans le diagramme :

- On ajoute la boucle évènement :

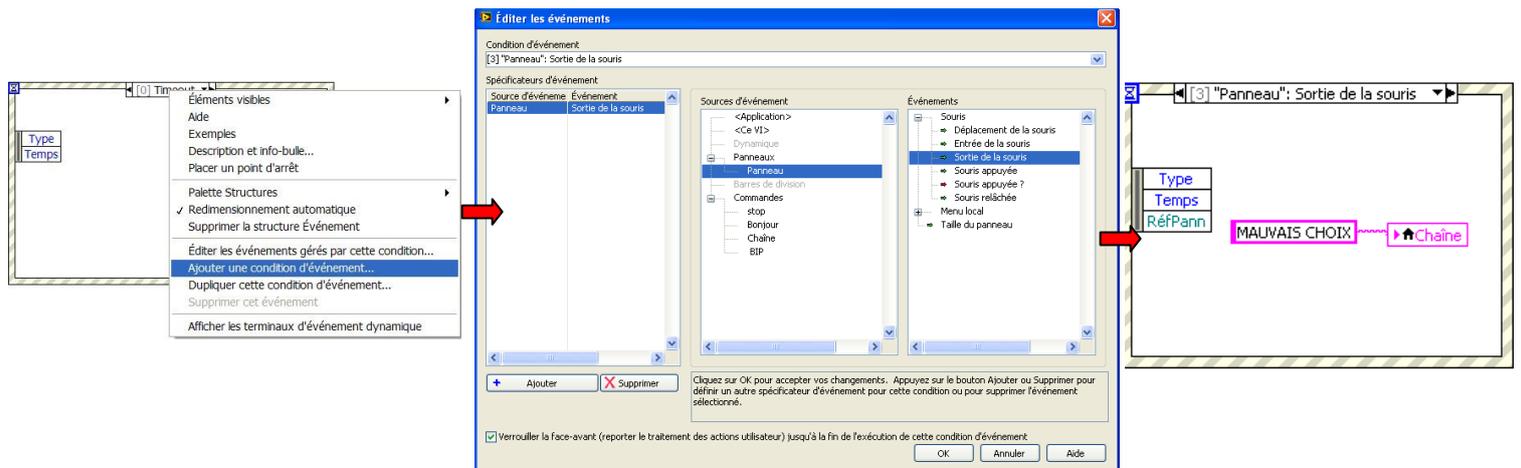
- On clique sur « Ajouter une condition évènement.... »
- Dans la colonne « Source évènement », on clique sur « BONJOUR »
- Dans la colonne « Evènement », on clique sur « Souris relâchée »
- Puis sur OK

On vient de créer l'évènement lorsqu'on cliquera sur le bouton BONJOUR de la face avant.

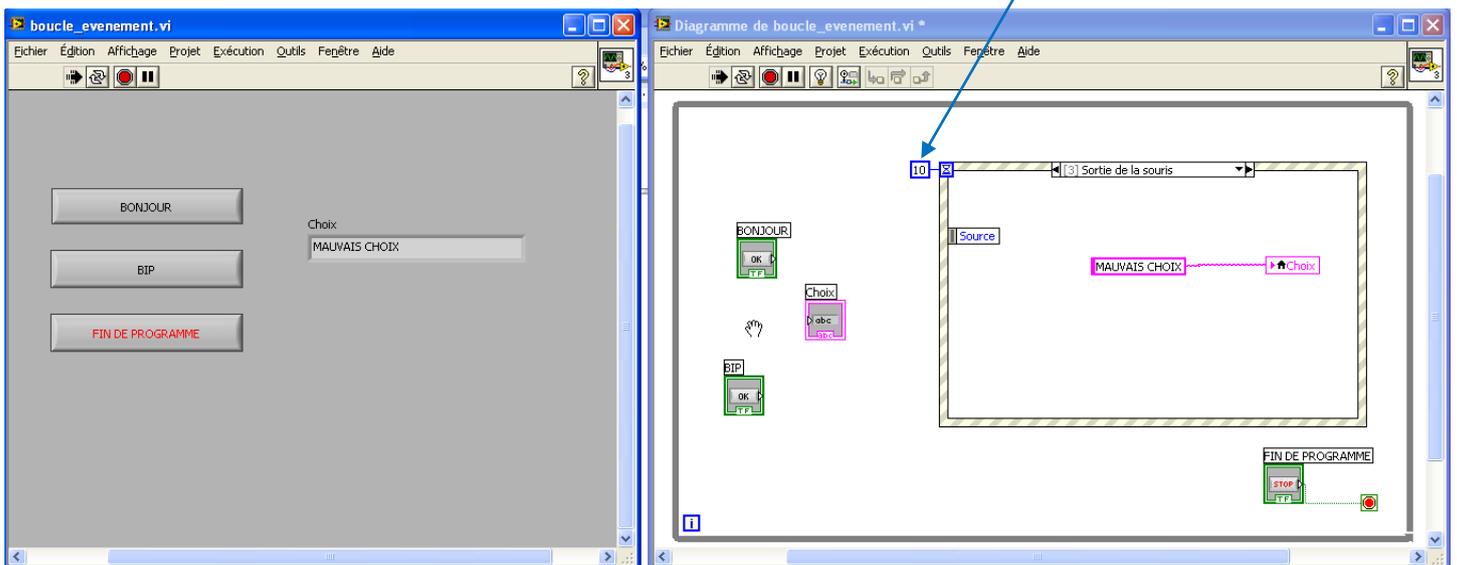
Dans le cadre, on créer une chaîne de caractère « BON CHOIX », une variable locale « CHOIX » et une boîte de dialogue avec la chaîne de caractère « BONJOUR ».



- De même on crée le second évènement : BIP
- On crée aussi l'évènement : MAUVAIS CHOIX



- On précise le temps de scrutation en ms, en cliquant sur le sablier avec le bouton droit et faire « Créer une constante » qui est à -1 par défaut et mettre 10 par exemple.
- On ajoute une structure de répétition
- On obtient :



e) Séquencement :

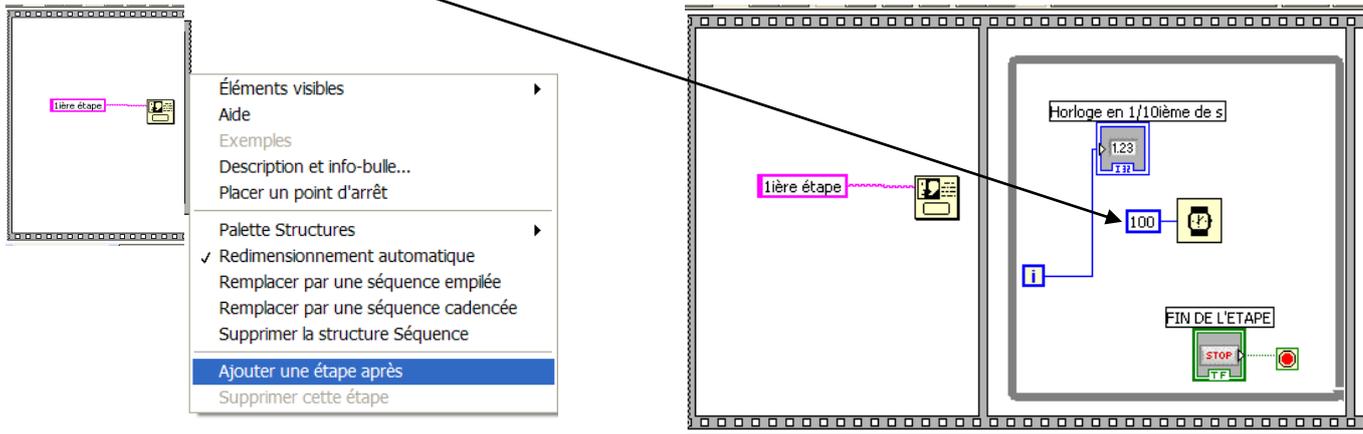
Lorsque l'on souhaite que des actions se déroulent dans un ordre bien défini, on utilise une structure de blocs séquence.

Exemple :

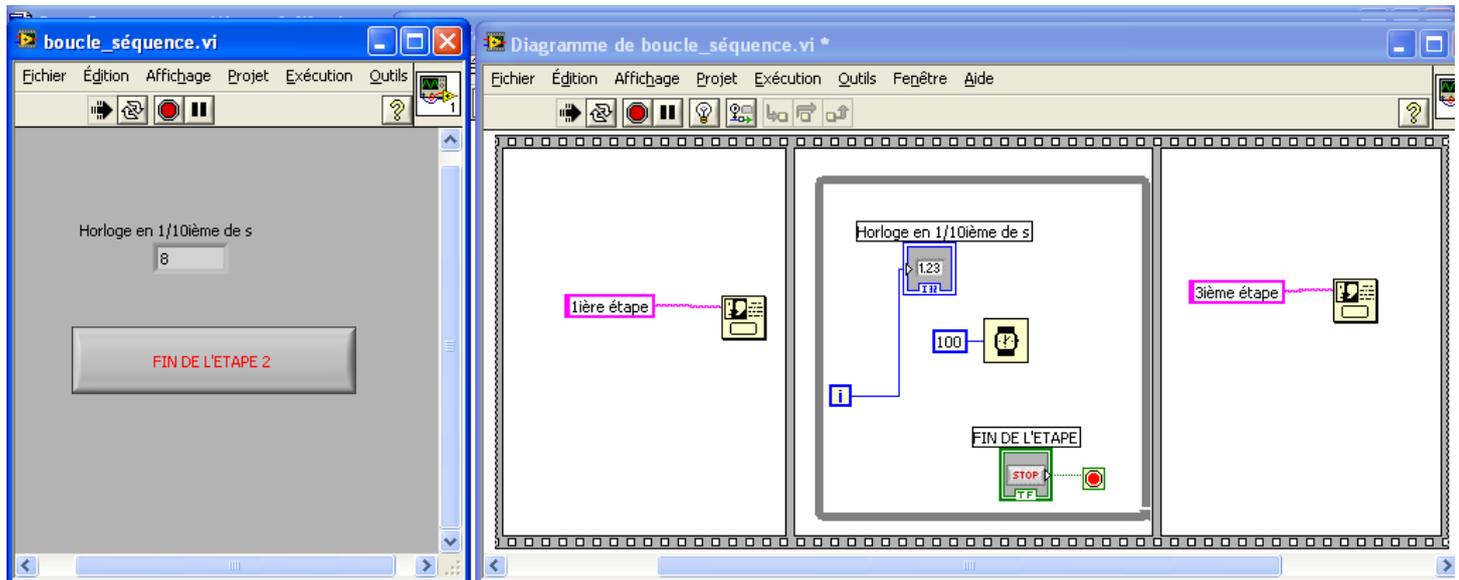
On va créer un programme qui :

- ❖ Affiche 1^{ière} séquence dans une boîte de dialogue,
- ❖ Puis qui affiche une horloge en 1/10^{ième} de seconde
- ❖ Puis qui affiche 3^{ième} séquence dans une boîte de dialogue.

- On crée un nouveau VI
- Sur la face avant on ajoute un indicateur numérique (Horloge en 1/10^{ème} de s)
- Dans le diagramme :
 - On ajoute la structure séquence déroulée
 - Dans cette étape on rajoute une boîte de dialogue avec la chaîne de caractère « 1^{ère} étape ».
 - On rajoute une étape en cliquant avec le bouton droit sur le bord de la structure et en choisissant « Ajouter une étape après »
 - On fait glisser l'indicateur numérique dans cette seconde étape et on ajoute un objet « attendre 100 ms » et une boucle while



- On rajoute troisième étape et une boîte de dialogue avec la chaîne de caractères « 3^{ème} étape ».
- On obtient :



Pour les programmes plus complexes, on peut superposer les séquences. On parle de séquences déroulées dans l'exemple ci-dessus et de séquences empilées dans le cas contraire.