

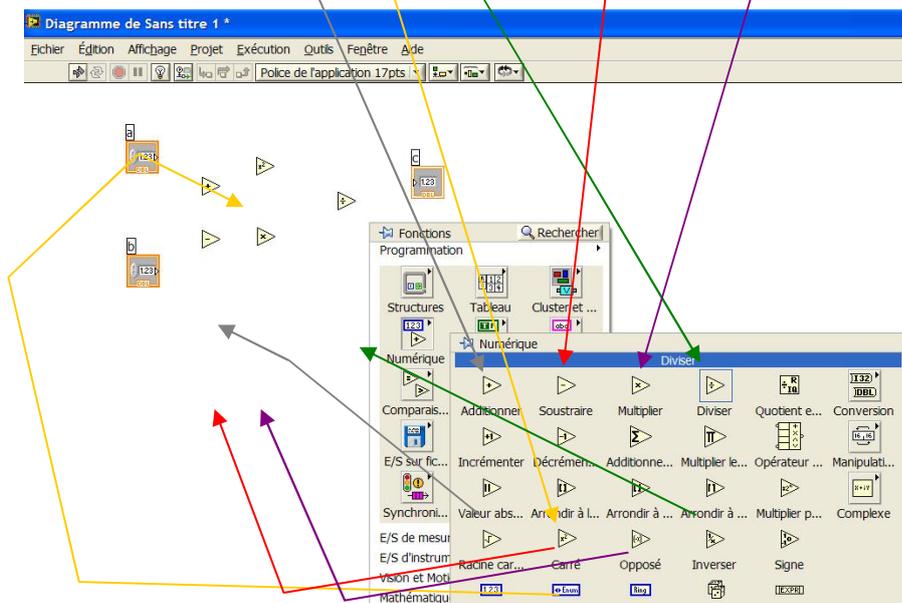
Initiation à LabView : Exercices supplémentaires :

A.1. Exercice 1 :

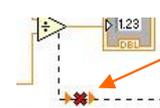
But du programme : calculer $c=(b+a)^2/((a-b)*(a+b))$

- Ouvrir le logiciel *LabView*
- Demander la création d'un nouveau projet vide .
- Ajouter un nouveau VI.
- Sur la face avant on ajoute deux commandes et un indicateur numériques (ils sont implicitement du type double) que l'on nomme a,b et c.
- Sur le diagramme :

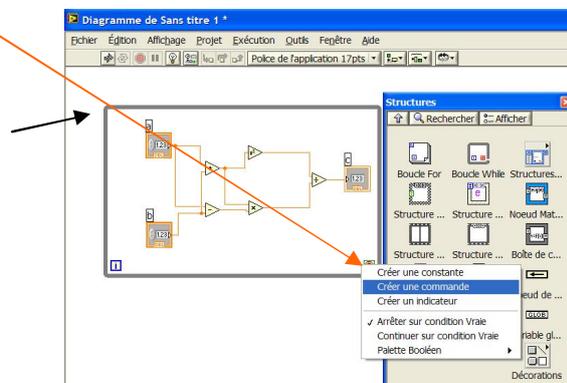
- on ajoute les opérateurs addition, carré, diviser, soustraire et multiplier.



- on câble sorties->entrées.... Sorties -> entrées. (Pour effacer un lien non relié (lien avec une croix rouge), on appui sur Ctrl et B en même temps)



- on ajoute une structure de répétition (on crée le bouton stop en cliquant sur « créer une commande » à l'aide du bouton droit sur le stop de la boucle while).

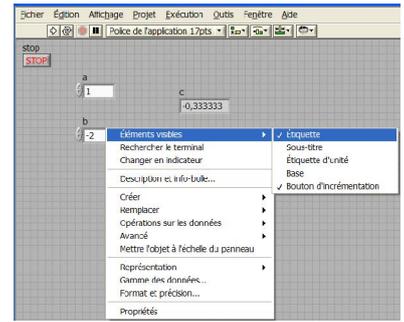


- Savegarde : enregistrer le projet et donner un nom au VI.
- Teste en exécution unique
- Fermer tout

Remarques : Modifications de l'interface utilisateur :

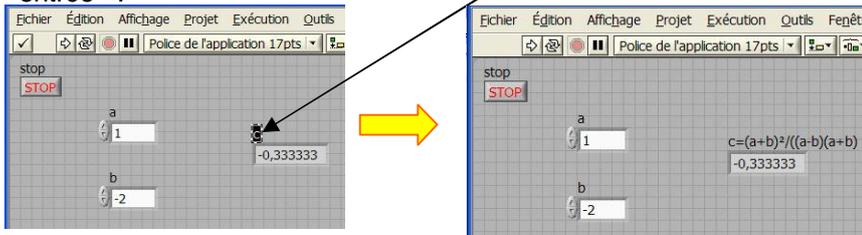
a) Les étiquettes :

Cet élément peut-être visible ou non : cliquer avec le bouton droit sur le rectangle de l'objet et choisir « **Éléments visibles** », « **étiquette** »

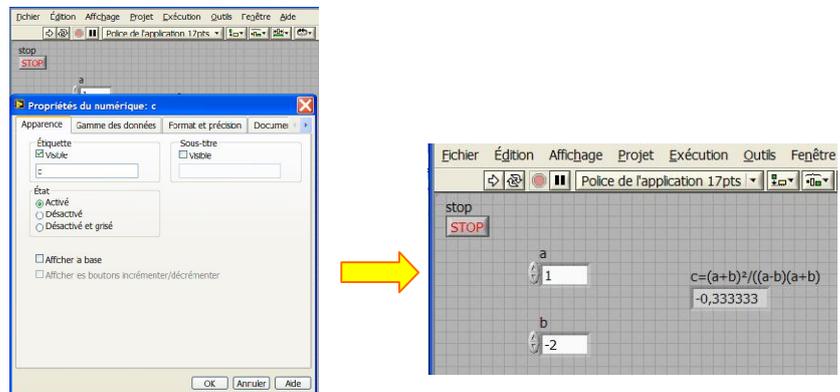


Pour modifier une étiquette, deux méthodes :

- un double clique sur l'étiquette, elle devient noire et vous tapez le nouveau texte et validez avec « **entrée** ».



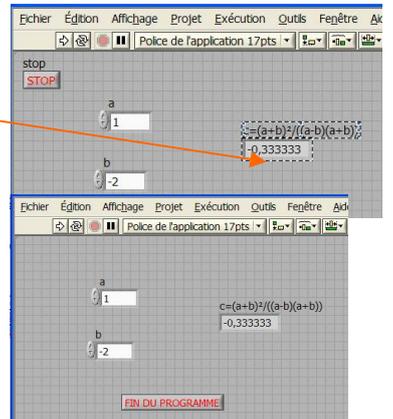
- cliquer avec le bouton droit sur le rectangle de l'objet et choisir « **Propriétés** » et modifier le texte puis « **OK** »



b) Position des objets :

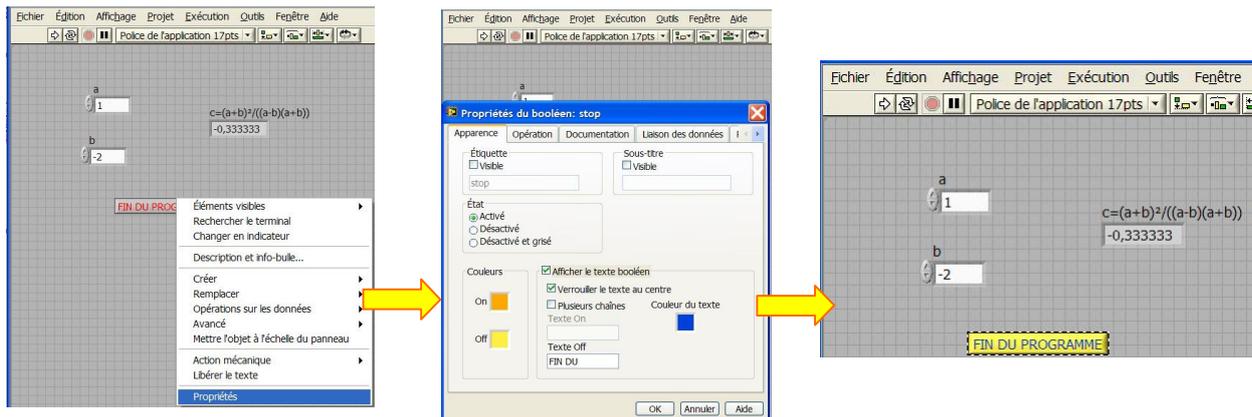
Si un objet est mal positionné, on peut le déplacer en cliquant dessus (il faut qu'il soit encadré de pointillé) et on le glisse à l'endroit souhaité en restant appuyé sur le bouton de souris.

Déplacer le bouton stop, cacher l'étiquette et remplacer le texte stop par « **FIN DE PROGRAMME** », vous devez obtenir la fenêtre suivante :

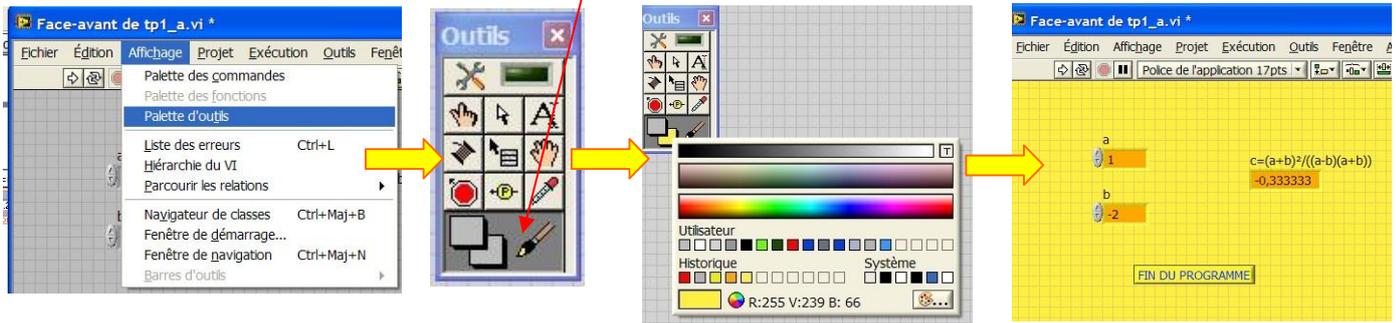


c) Les couleurs :

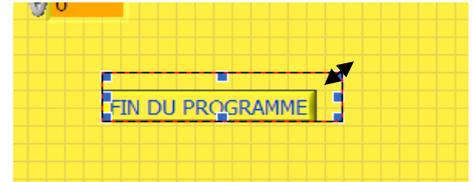
Pour modifier les couleurs du bouton « **FIN DE PROGRAMME** », cliquer sur l'objet avec le bouton droit et sélectionner « **Propriété** » puis dans la fenêtre « **apparence** » mettre les couleurs souhaitées et



Pour modifier la couleur de l'arrière plan ou celle des indicateurs et commandes, cliquer sur « Affichage », « Palette outils » et choisir le pinceau, la couleur souhaitée puis l'endroit à colorier.



- d) Les dimensions :
Pour redimensionner un objet, il suffit de cliquer et déplacer les carrés qui définissent les dimensions de l'objet.

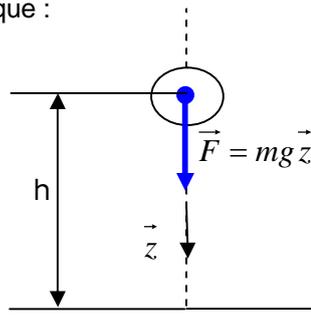


A.2. Exercice 2 :

BUT : vitesse au contact du sol d'un projectile :

On désire créer un programme qui calcule la vitesse d'un projectile, lancé d'une hauteur h (à donner par l'utilisateur) avec vitesse initiale nulle, lorsqu'il entre en contact avec le sol.

Rappel de mécanique :



On applique de principe fondamental de la statique $\sum \vec{F} = m\vec{a}$ d'où suivant \vec{z} :

$$a = g$$

$$v = gt + v_0$$

$$x = \frac{gt^2}{2} + v_0t + x_0$$

Or à $t=0$: $v = 0$ et $x = -h$ d'où $v_0 = 0$ et $x_0 = -h$ donc

$$a = g$$

$$v = gt$$

$$x = \frac{gt^2}{2} - h$$

Lorsque le projectile touche le sol, $x=0$ donc $t = \sqrt{\frac{2h}{g}}$ d'où $v = \sqrt{2gh}$.

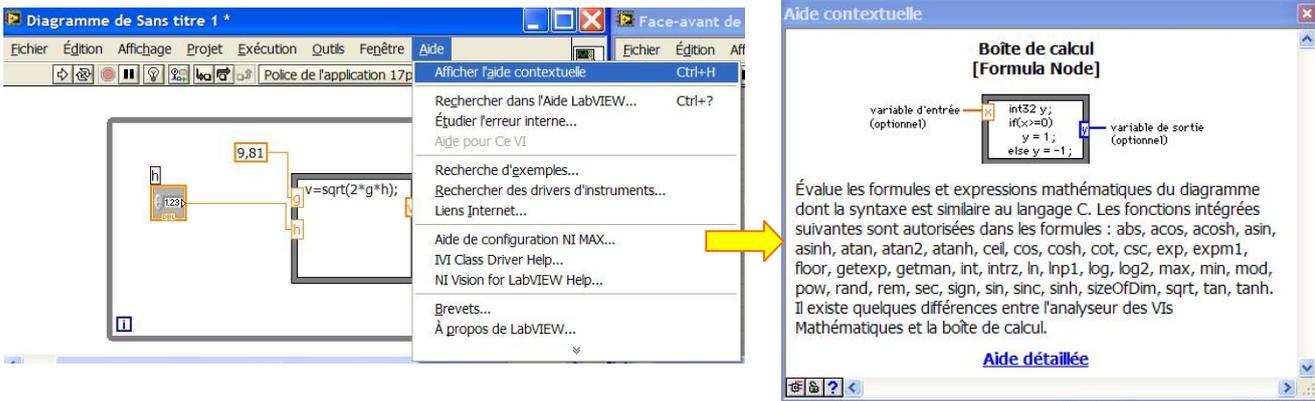
On utilisera une boîte de calcul pour faire le programme.

- Ouvrir le logiciel *LabView*
- Demander la création d'un nouveau projet vide.
- Ajouter un nouveau VI.
- Sur la face avant, on ajoute une commande (Hauteur de lancement en m) et un indicateur numériques (vitesse au sol en m/s).

e) Sur le diagramme :

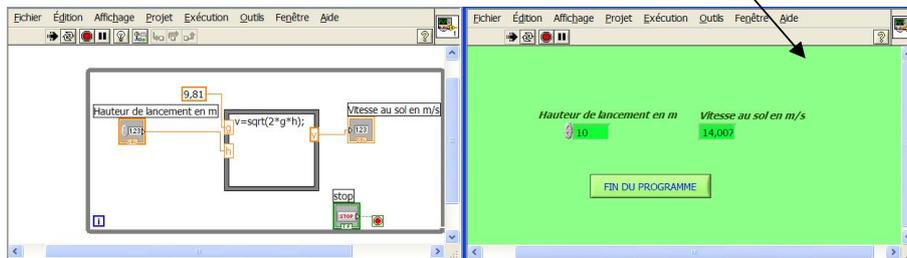
- ✚ ajouter une boîte de calcul et une constante g (donner la valeur 9,81).
- ✚ créer deux variables d'entrée h et g en cliquant avec le bouton droit sur le bord gauche de la boîte de calcul.
- ✚ créer une variable de sortie v en cliquant avec le bouton droit sur le bord droit de la boîte de calcul.
- ✚ lier les variables d'entrée aux commandes et la sortie à l'indicateur.
- ✚ éditer la formule de calcul dans la boîte de calcul. ($\sqrt{\quad}$ s'écrit sqrt dans la boîte de dialogue)

Remarque : les différents sigles peuvent être vu en cliquant sur « aide » « afficher l'aide contextuelle » et en allant sur l'objet concerné.



- ✚ Créer la boucle while avec sa commande « fin de programme »

- f) Modifier l'interface utilisateur pour obtenir l'image ci-dessous.
- g) Sauvegarde : enregistrer le projet et donner un nom au VI.
- h) Teste en exécution unique
- i) Fermer tout



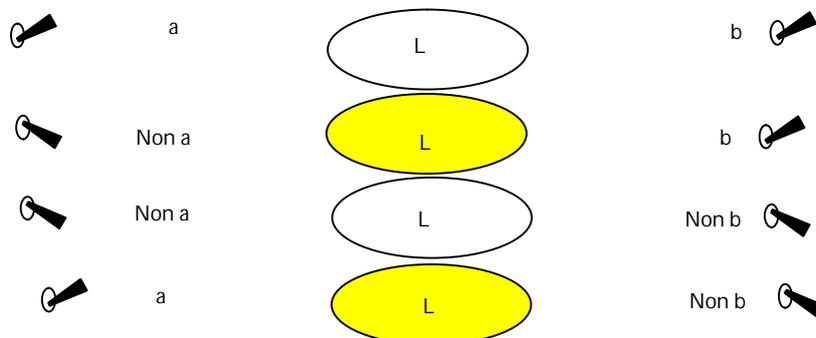
A.3. Exercice 3 :

BUT : Simulation d'une équation logique : va et vient d'un escalier:

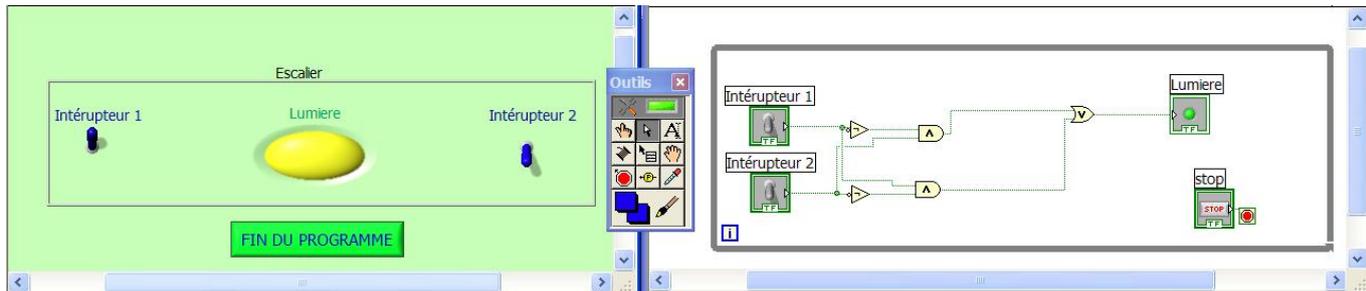
On se propose de réaliser le câblage d'un système d'éclairage en deux points différents d'un escalier avec deux interrupteurs.

Soient les interrupteurs a et b et une lampe L

Fonctionnement : Une action sur a ou b provoque le changement d'état de la lampe.



- a) Demander la création d'un nouveau projet vide.
- b) Ajouter un nouveau VI.
- c) Sur la face avant :
 - ✚ Ajouter les commandes booléennes a et b (interrupteurs)
 - ✚ Ajouter un indicateur booléen L (lumière)
 - ✚ Ajouter un bouton « FIN DE PROGRAMME »
- d) Sur le diagramme :
 - ✚ Insérer les fonctions ET, NON et OU
 - ✚ Câbler l'équation $L = a \cdot b + a \cdot b$
 - ✚ Ajouter une boucle while
- e) Sauvegarde : enregistrer le projet et donner un nom au VI.
- f) Teste en exécution unique
- g) Fermer tout



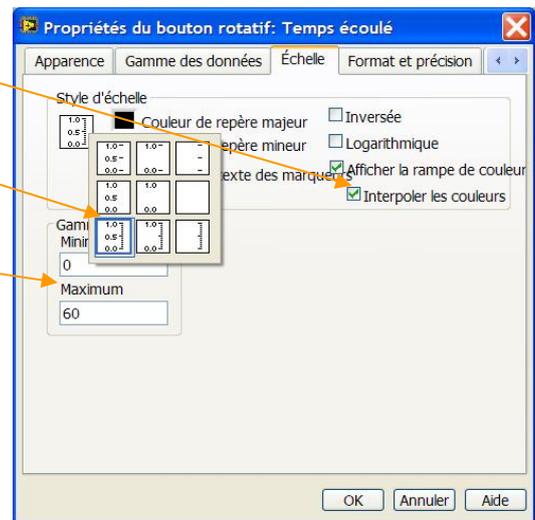
A.4. Exercice 4 :

BUT : Affichage d'un chronomètre:

On se propose de réaliser un programme qui représente un chronomètre qui démarre au lancement du programme et s'arrête automatiquement à la fin de la durée donnée par l'utilisateur (en seconde $0 < t < 60$).

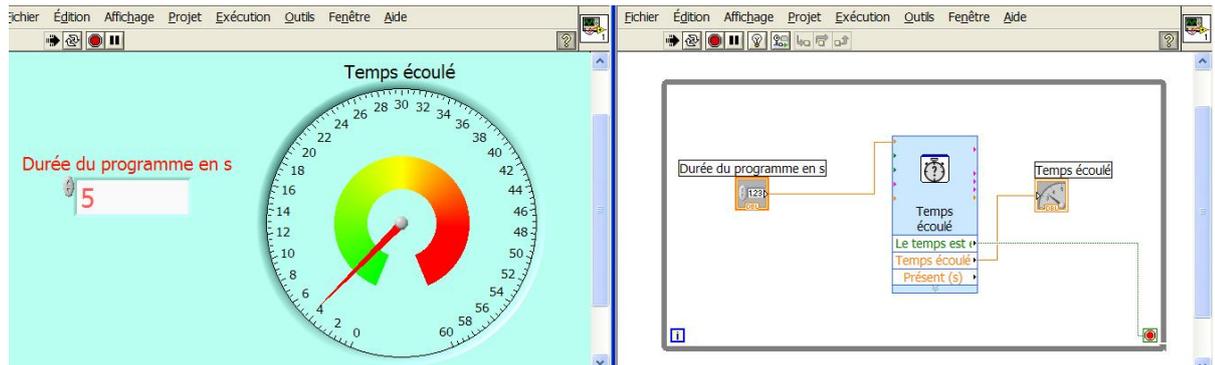
- a) Demander la création d'un nouveau projet vide.
- b) Ajouter un nouveau VI.
- c) Sur la face avant :
 - ✚ Ajouter une commande numérique nommée « Durée du programme en s »
 - ✚ Ajouter une jauge nommée « temps écoulé »
 - ✚ Modifier les propriétés de la jauge : sur la page échelle :

- Cocher afficher la rampe de couleur
- Choisir le style de l'échelle ci-contre
- Gamme : Maximum : taper 60



- ✚ Agrandir la jauge
- d) Sur le diagramme :
 - ✚ Insérer une information temporelle : temps écoulé
 - ✚ Ajouter une boucle while
 - ✚ Câbler les éléments (le stop de la boucle est câblé à la variable booléenne : le temps est écoulé)
 - e) Sauvegarde : enregistrer le projet et donner un nom au VI.

- f) Teste en exécution unique
- g) Fermer tout



A.5. Exercice 5 :

BUT : Trouver les racines du polynôme du 2° degré ax^2+bx+c . On envisage deux résolutions, soit dans le domaine des réels, soit sur les complexes:

L'utilisateur saisit les coefficients a, b, c.

En cliquant sur un bouton « Calculer RACINES », il provoque le calcul et l'affichage de la valeur des racines ou un message indiquant qu'il n'y en a pas

En cliquant sur le bouton « Calculer DELTA », il provoque l'affichage de delta

Cas de la solution dans le domaine des réels :

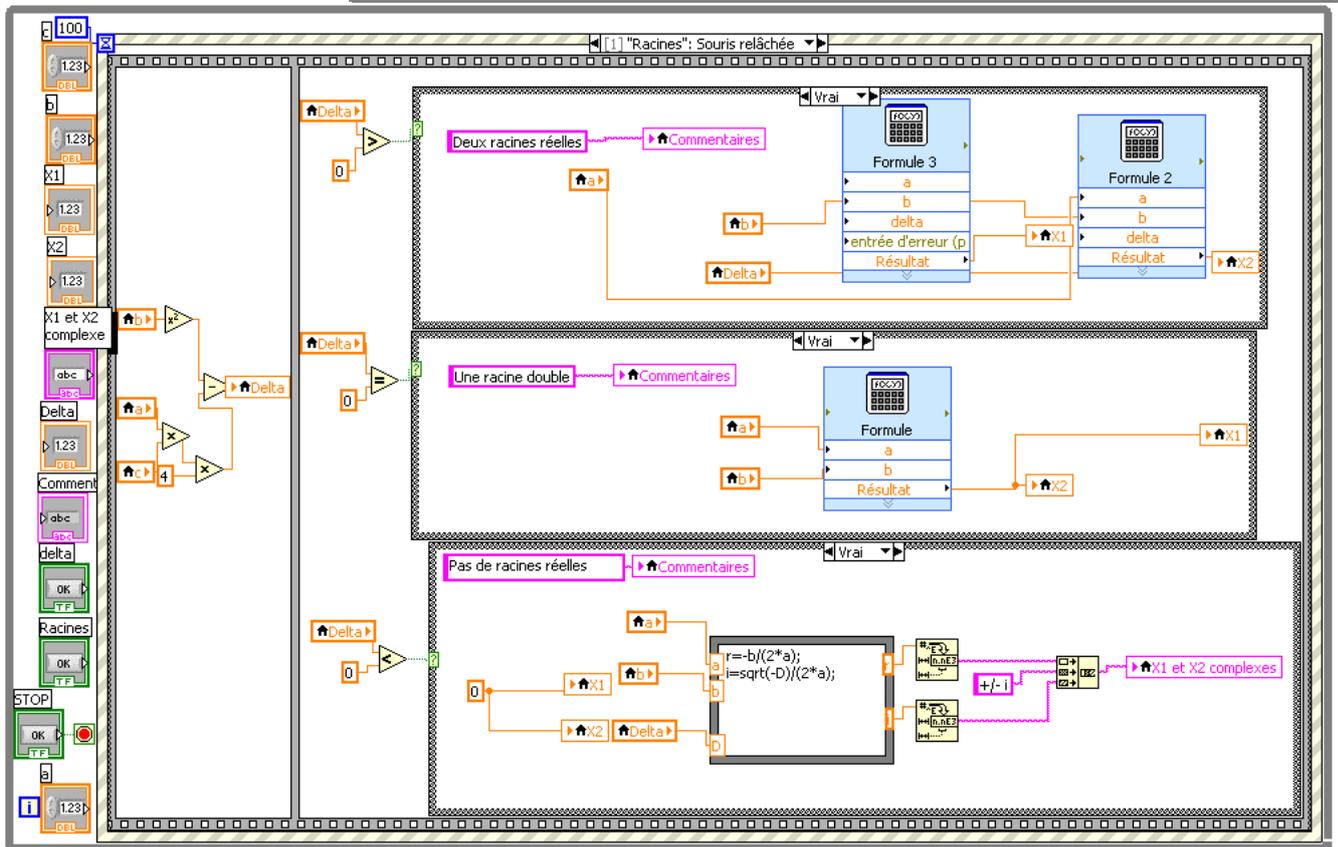
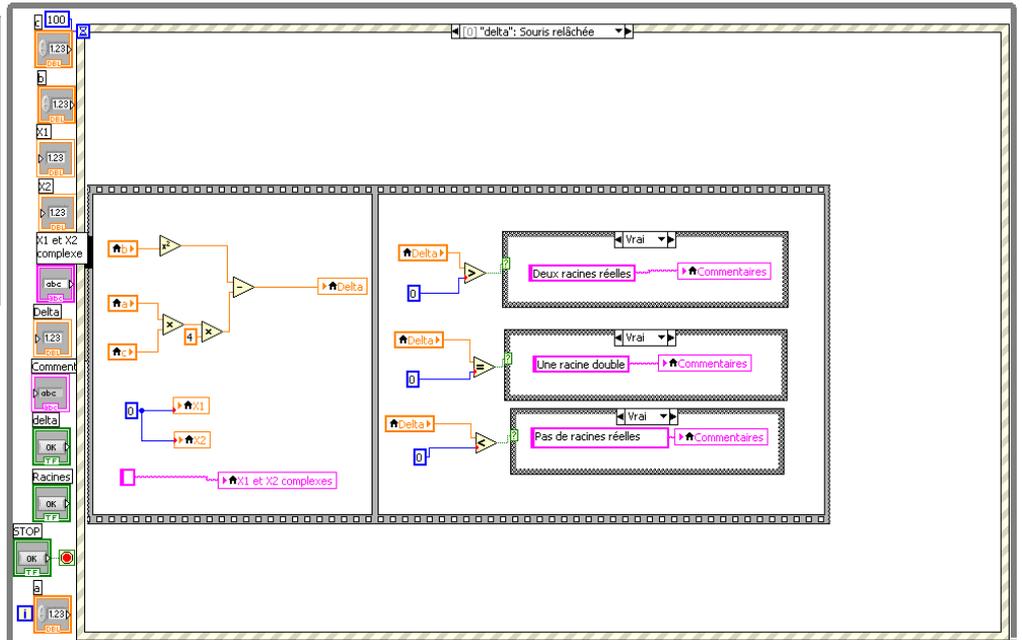
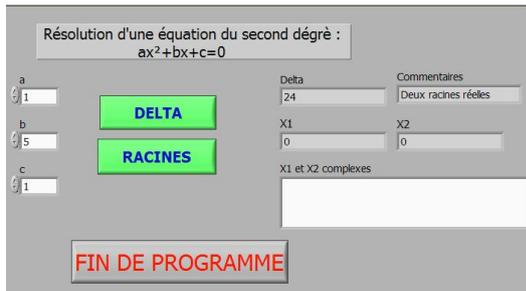
- + Saisie des valeurs
- + Calculer DELTA =
- + Si L'utilisateur choisi Calculer RACINES : traiter les cas :
 - Si $\Delta > 0$ $x_1 = \dots$ $x_2 = \dots$ Message = « Deux racines réelles »
 - Si $\Delta = 0$ $x_1 = x_2 = \dots$ Message = « Une racine double »
 - Si $\Delta < 0$ Message = « Pas de racines réelles »

Cas de la solution dans le domaine des complexes :

- + Le traitement est le même pour $D \geq 0$.
- + Pour $D < 0$, il faut calculer la partie réelle des racines $-\frac{b}{2 \cdot a}$ et la partie imaginaire $\frac{\sqrt{\Delta}}{2 \cdot a}$ et faire écrire les deux parties du résultat.

- a) Demander la création d'un nouveau projet vide.
- b) Ajouter un nouveau VI.
- c) Sur la face avant :
 - + Ajouter les commandes « a », « b » et « c » de type réels
 - + Ajouter les indicateurs numérique « x1 », « x2 » et « delta », un indicateur chaîne de caractères « commentaires »
 - + Trois boutons : « DELTA », « RACINES » et « FIN DE PROGRAMME »
- d) Sur le diagramme :
 - + Câbler le calcul de Δ
 - + Ajouter les comparaisons $\Delta > 0$, $\Delta = 0$ et $\Delta < 0$
 - Pour chaque cas créer une structure condition
 - Dans chaque condition ajouter des variables locales x1, x2, Commentaires
 - Pour chaque cas faire calculer x1, x2 et créer la constante permettant de donner le commentaire correct.
 - + Ajouter une boucle while
 - + Câbler les éléments
- e) Sauvegarde : enregistrer le projet et donner un nom au VI.
- f) Teste en exécution unique

g) Fermer tout



A.6. Exercice 6 :

BUT : Trouver un nombre aléatoire compris entre 1 et 100 choisi par l'ordinateur en 10 coups maximum

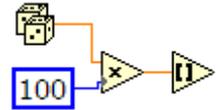
- ✦ L'utilisateur démarre la partie en cliquant sur le bouton « JOUER »
- ✦ L'utilisateur saisit un nombre et valide son essai en cliquant sur le bouton « ESSAI ».
- ✦ L'utilisateur gagne s'il trouve avant 10 essais.

- ✦ Dans la case message, on affiche « Pour démarrer cliquez sur JOUER »
- ✦ Démarrage du jeu en cliquant sur « JOUER » et un message s'affiche : jeu en cours.
- ✦ Saisi d'un nombre
- ✦ L'utilisateur clique sur « ESSAI »

- Si le nombre proposé est trop petit, le bouton « Trop petit » s'allume et l'indicateur « Nb d'essai » s'incrémente de 1, puis si « Nb d'essai » < 10 on continue sinon on ouvre une boîte de dialogue « vous avez perdu, le nombre recherché était ??? »
- Si le nombre proposé est trop grand, le bouton « Trop grand » s'allume et l'indicateur « Nb d'essai » s'incrémente de 1, puis si « Nb d'essai » < 10 on continue sinon on ouvre une boîte de dialogue « vous avez perdu, le nombre recherché était ??? »
- Si le nombre proposé est la valeur cherché, le bouton « Trouvé » s'allume, une boîte de dialogue s'ouvre et affiche « Vous avez trouvé en ??? coups » et la partie se termine.

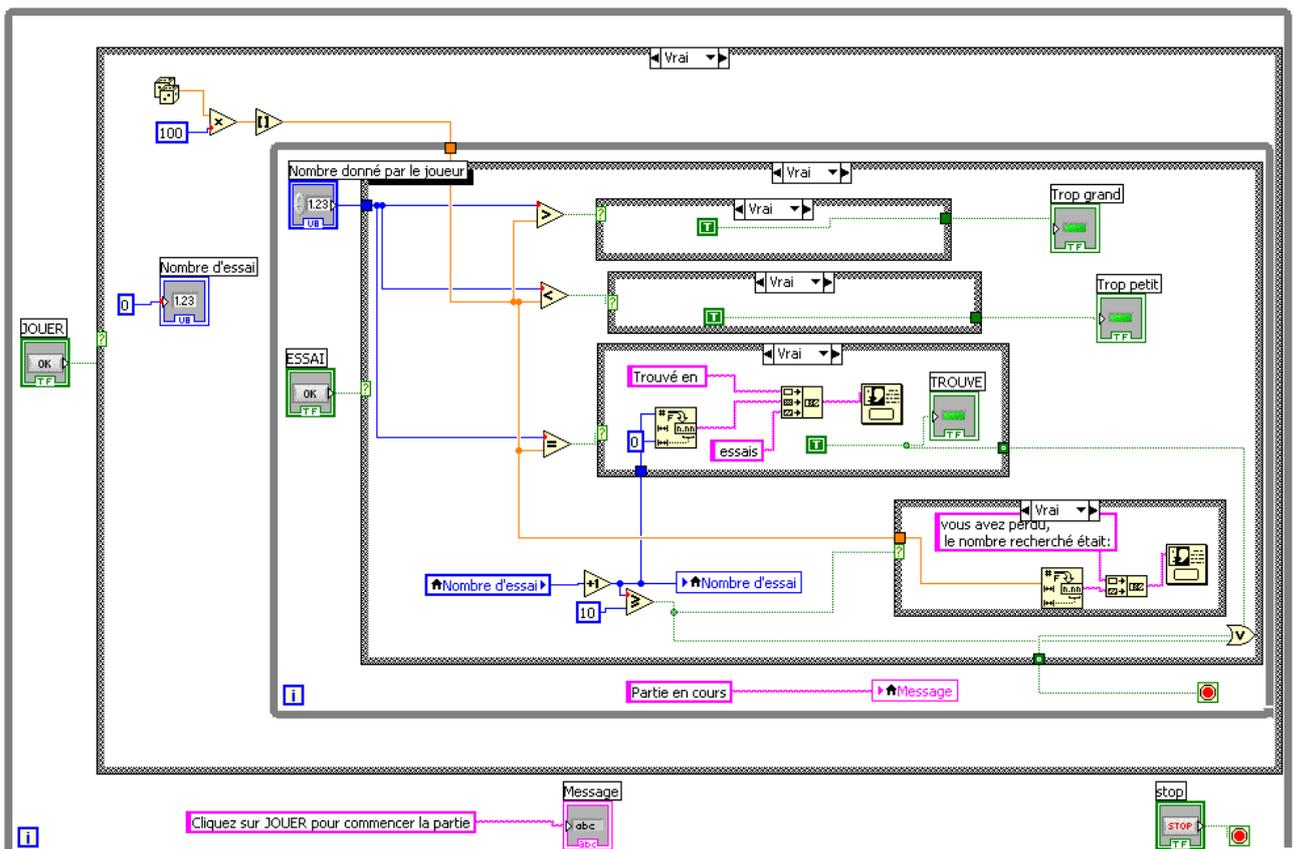
- a) Demander la création d'un nouveau projet vide.
- b) Ajouter un nouveau VI.
- c) Sur la face avant :
 - ✚ Ajouter la commande « Nombre donné par le joueur » du type entier.
 - ✚ Ajouter un indicateur numérique « Nombre d'essai », un indicateur chaîne de caractère « Message »
 - ✚ Trois boutons : ESSAI, JOUER et FIN DE PROGRAMME.
 - ✚ Trois voyants : TROUVE, TROP PETIT et TROP GRAND.

- d) Sur le diagramme :
 - ✚ Créer la boucle While principal qui se termine si on clique sur « FIN DE PROGRAMME »
 - ✚ Créer le message « Cliquez sur JOUER pour démarrer la partie »
 - ✚ Créer la structure condition si on clique sur « JOUER »
 - ✚ Initialiser le chiffre « Nombre d'essai »
 - ✚ Créer un nombre aléatoire (entre 0-1 que l'on va multiplier par 100 et prendre la partie entière)



Créer la structure tant que Nombre essai < 10 ou Nombre trouvé faux on continue la partie et on affiche « partie en cours »

- e) Créer la structure condition lorsqu'on clique sur « ESSAI »
- f) Créer les trois cas : trop petit, trop grand et trouvé.
- e) Sauvegarde : enregistrer le projet et donner un nom au VI.
- f) Teste en exécution unique
- g) Fermer tout



A.7. Exercice 7 :

BUT : On souhaite gérer un tableau de mesures. Ces mesures sont des nombres réels tous positifs. Le programme est exécuté tant que l'opérateur ne clique pas « FIN DE PROGRAMME »

L'utilisateur doit pouvoir :

- A. Initialiser le tableau, c'est dire le définir comme n'ayant aucun élément
- B. Saisir une valeur et pouvoir :
 - o La ranger en début de tableau (indice 0)
 - o Ou la ranger en fin de tableau
 - o Ou à une position définie par lui
- C. Supprimer une valeur à une position définie par lui même
- D. Calculer la moyenne des valeurs
- E. Faire tracer les valeurs en fonction de leur rang dans le tableau
- F. Supprimer l'offset des valeurs (valeur moyenne)
- G. Classer les valeurs
- H. Trouver le maxi et le mini du tableau

a) Demander la création d'un nouveau projet vide.

b) Ajouter un nouveau VI.

c) Sur la face avant :

✚ Créer un tableau et ajouter dedans un indicateur du type désiré (numérique double)

✚ On ajoute ensuite une série de boutons correspondant aux fonctions définies dans le cahier des charges :

- | | |
|---------------------------------|-----------------------|
| ⇒ Initialiser | ⇒ Moyenne |
| ⇒ Insérer en tête | ⇒ Tracer |
| ⇒ Insérer en fin | ⇒ Supprimer l'offset |
| ⇒ Insérer au rang n | ⇒ Classer les valeurs |
| ⇒ Supprimer la valeur de rang n | ⇒ Max Min |

✚ On crée ensuite des commandes numériques (entrées de l'utilisateur) permettant de définir :

- ⇒ La valeur à insérer
- ⇒ La place d'insertion ou de suppression

✚ Trois indicateurs permettant d'afficher :

- o La valeur moyenne
- o Le mini
- o Le maxi

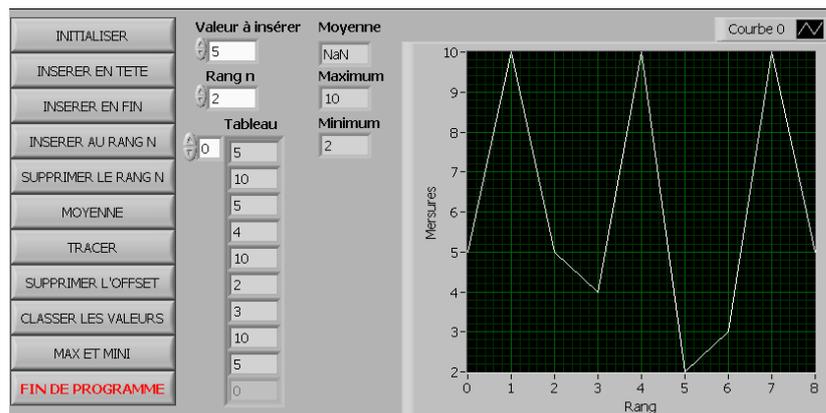
✚ On utilise le graphe « Graphe » qui affiche un tableau de valeurs en fonction de leur rang

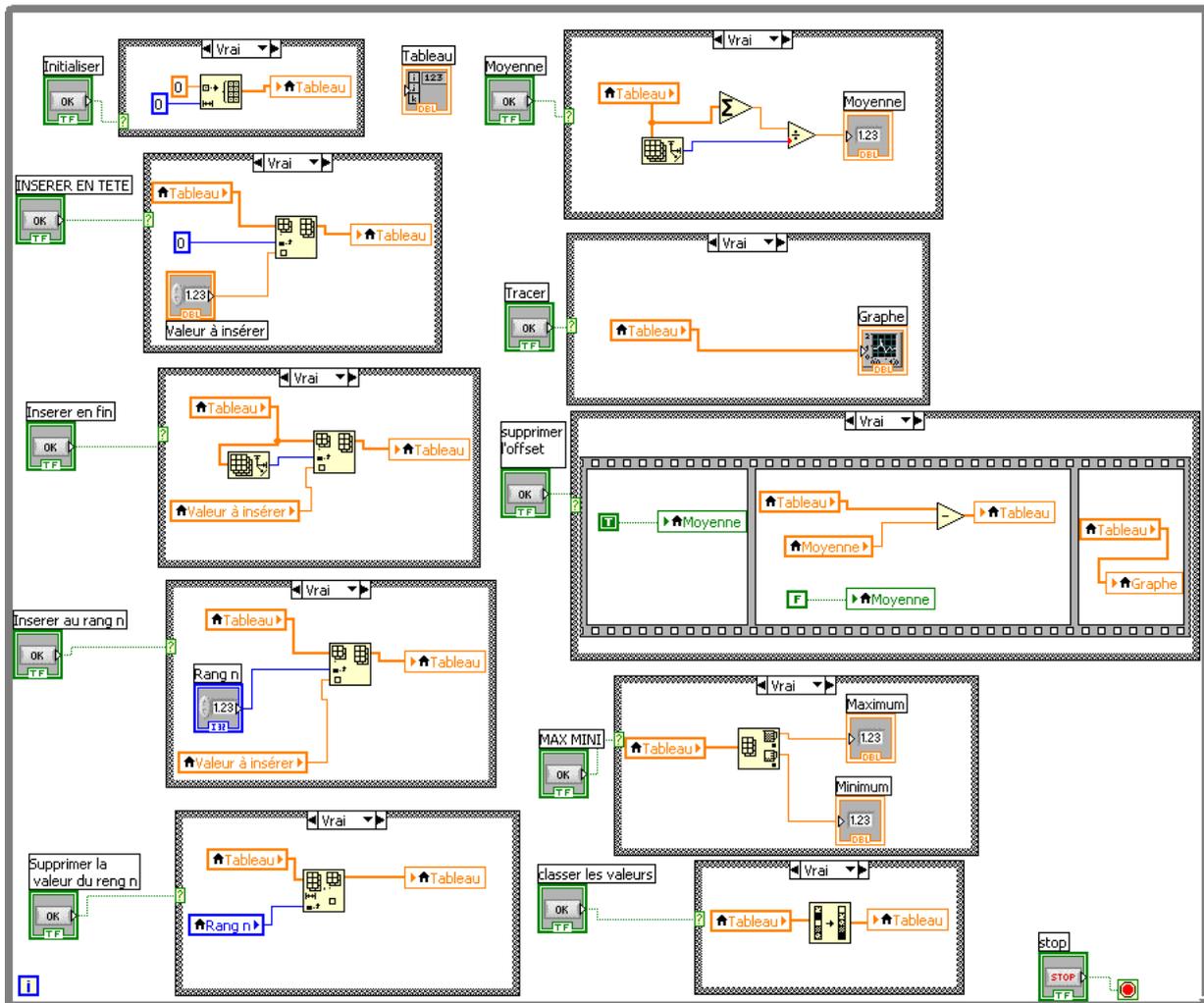
d) Sur le diagramme :

Créer une condition SI par tâche à réaliser

✚ Pour chaque condition :

- o Une variable locale tableau en lecture
- o Une variable locale tableau en écriture
- o La fonction à réaliser par la condition

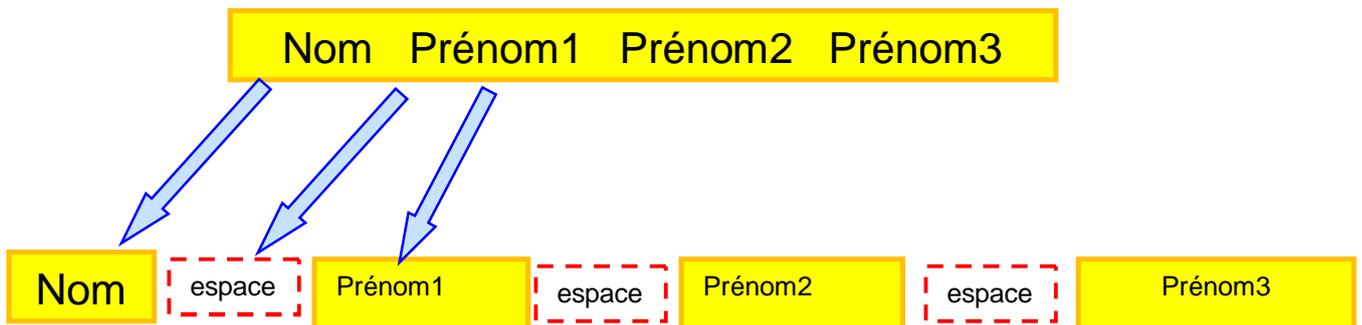




- e) Sauvegarde : enregistrer le projet et donner un nom au VI.
- f) Teste en exécution unique
- g) Fermer tout

A.8. Exercice 8 :

BUT : Effectuer des traitements de chaînes de caractères pour en extraire des informations élémentaires.



On veut que le logiciel remplisse quatre indicateurs de chaîne avec les informations extraites de la chaîne principale (tapée par l'utilisateur) qui sera stockée dans une commande de chaînes.

Dès que l'utilisateur a tapé sa chaîne de caractères, il clique sur le bouton TRAITER, chaque indicateur doit se remplir.

Si l'opérateur n'a saisi qu'un prénom, une boîte de dialogue s'ouvre pour demander confirmation :

- s'il répond OUI, les indicateurs Prénom 2 et Prénom 3 affiche « PAS DE DEUXIEME PRENOM » et « PAS DE TROISIEME PRENOM »
- s'il répond NON, un message s'affiche dans une boîte, « veuillez rajouter votre second et troisième prénoms dans la chaîne principale puis appuyer sur traiter ».

Si l'opérateur n'a saisi que deux prénoms, une boîte de dialogue s'ouvre pour demander confirmation :

- s'il répond OUI, l'indicateur Prénom 3 affiche « PAS DE TROISIEME PRENOM »
- s'il répond NON, un message s'affiche dans une boîte, « veuillez rajouter votre troisième prénoms dans la chaîne principale puis appuyer sur traiter ».

- Demander la création d'un nouveau projet vide.
- Ajouter un nouveau VI.
- Sur la face avant :
 - ✚ Créer 2 boutons FIN DE PROGRAMME et TRAITER.
 - ✚ Créer 4 indicateurs de chaîne pour chaque élément
 - ✚ Créer 1 commande pour la chaîne principale
- Sur le diagramme :
 - ✚ Exécuter le programme tant que fin de programme n'est pas relâché.
 - ✚ Si évènement bouton TRAITER relâché alors extraire les valeurs.

Veillez entrer Votre Nom puis Prénom1 puis Prénom2 puis Prénom3 séparés d'un espace puis appuyer sur TRAITER

DUPONT SERGE NICOLAS

NOM

DUPONT

TRAITER

PRENOM 1

SERGE

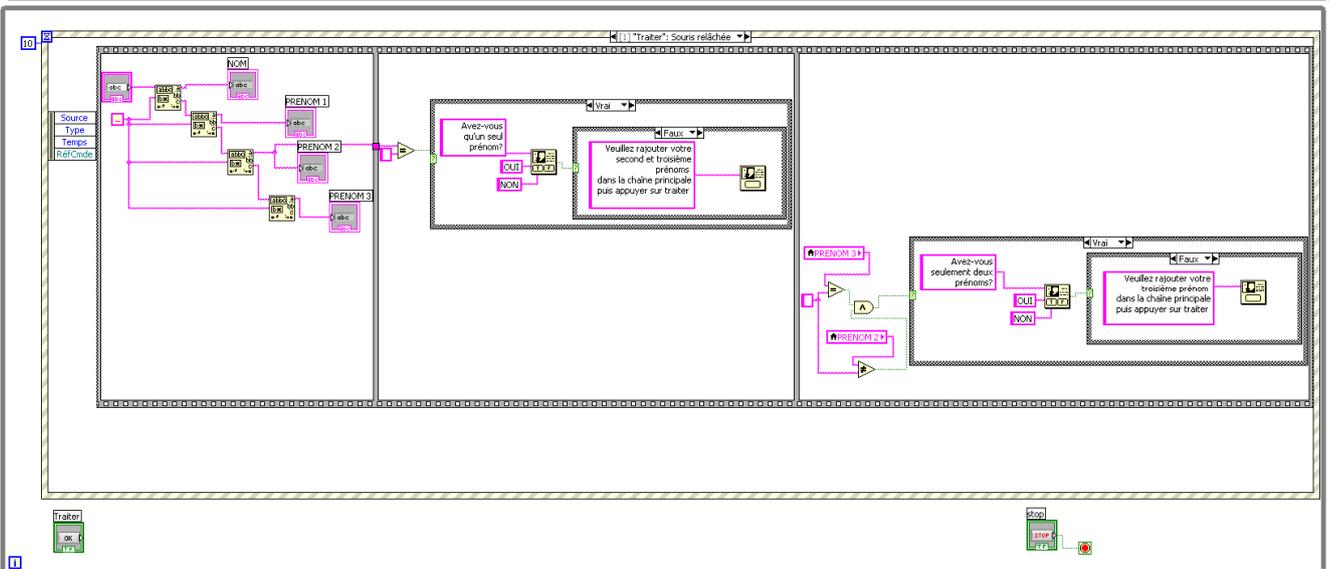
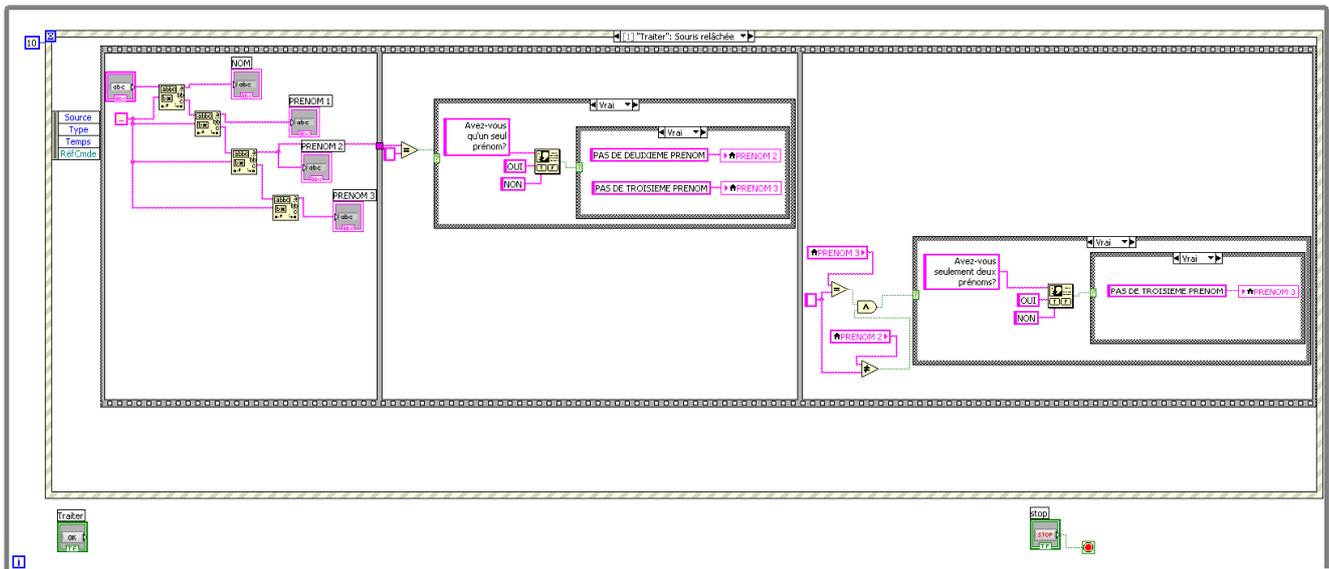
FIN DE PROGRAMME

PRENOM 2

NICOLAS

PRENOM 3

PAS DE TROISIEME PRENOM



- Sauvegarde : enregistrer le projet et donner un nom au VI.
- Teste en exécution unique
- Fermer tout