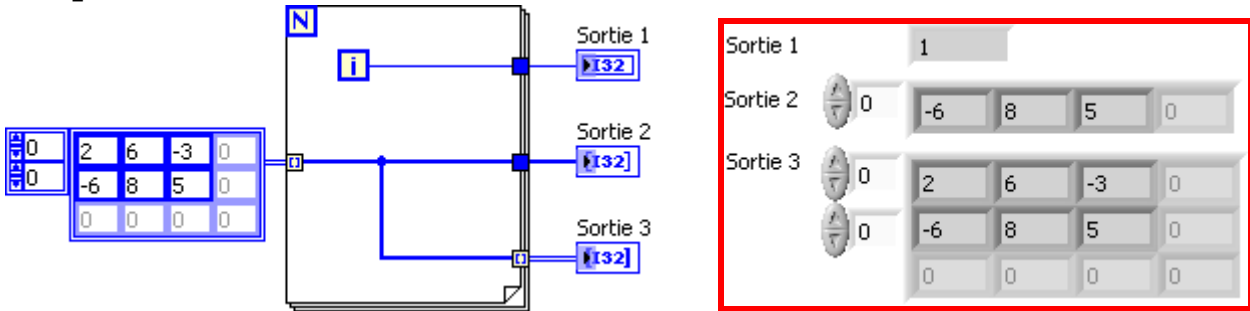
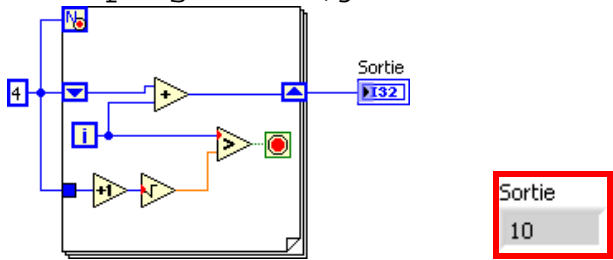


NOM :	<b>Correction</b> <b>Examen Final IV50</b>	Note : <div style="border: 1px solid black; padding: 2px; display: inline-block;">/20</div>
Durée : 1h. Sans document. PC, calculatrice et Téléphone portable interdits.		

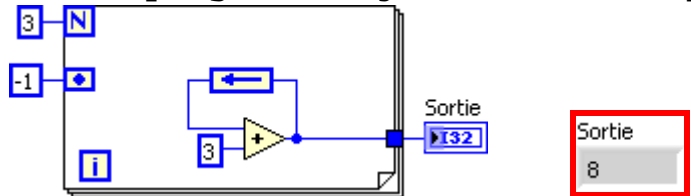
1,5 1) Dans le programme suivant donnez la valeur des sorties 1, 2 et 3 après exécution du programme (justifiez la réponse).



1,5 2) Dans le programme suivant donnez la valeur de la sortie après exécution du programme (justifiez la réponse).



1,5 3) Dans le programme suivant donnez la valeur de la sortie après exécution du programme (justifiez la réponse).

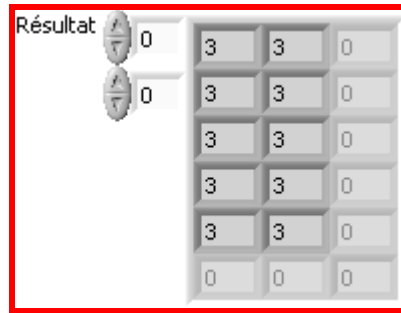
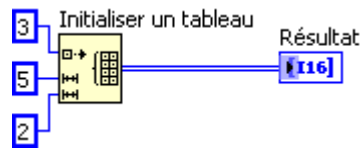


1,5 4) Expliquez la différence entre un événement de type filtre et un événement de type notification.

Les événements de type **notification** indiquent qu'une action utilisateur a déjà eu lieu, comme par exemple le changement de la valeur d'une commande. Utilisez les événements de type notification pour répondre à un événement après qu'il a eu lieu et que LabVIEW l'a traité

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. Utilisez les événements de type filtre pour participer à la gestion de l'événement, en écrasant éventuellement le comportement par défaut de l'événement. Les événements de type filtre possèdent des noms qui se terminent par un point d'interrogation

- 1) 5) Dans le programme suivant donnez la valeur de l'indicateur résultat après exécution du programme (justifiez la réponse).



- 3) 6) Complétez le tableau suivant.

	Variable Locale	Variable Globale	Variable partagée
Liée à un objet de face avant	Toujours	Oui mais FA du VI variable globale	Pas nécessairement
Echange de données dans un même VI	OUI	OUI	OUI
Echange de données entre plusieurs VI sur une même machine.	NON	OUI	OUI
Echange de données entre plusieurs VI sur des machines différentes.	NON	NON	OUI

- 2) 7) Ces éléments peuvent-ils transférer des données ? Complétez le tableau.

	OUI	NON
Les files d'attente	x	
Les sémaphores		x
Les variables locales	x	
Les notificateurs	x	

- 2) 8) Vous personnalisez une commande et vous la sauvegardez dans le fichier IV50.ctl. Vous utilisez ensuite cette commande personnalisée dans plusieurs VI. Vous ouvrez le fichier IV50.ctl, vous modifiez à nouveau cette commande et vous la sauvegardez sous le même nom, sera-t-elle modifiée automatiquement dans les VI qui l'utilisent ?

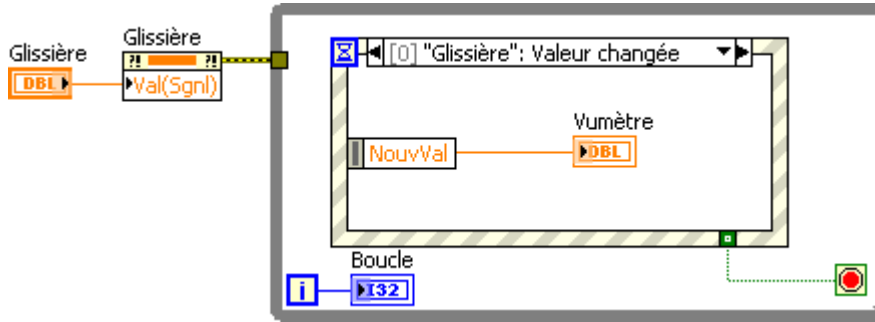
Si oui, expliquez le fonctionnement.

Si non, expliquez comment il faudrait faire.

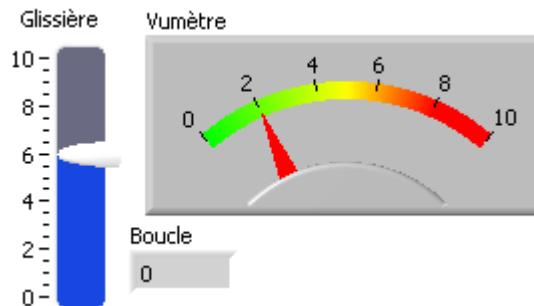
NON. L'utilisation d'une commande personnalisée sauvegardée dans un fichier .ctl créé en fait une copie de la commande dans le VI concerné. Une modification du fichier .ctl n'affecte pas la copie.

Pour qu'une mise à jour des modifications s'effectue dans tous les VI qui utilisent une commande personnalisée, il est nécessaire que cette commande soit sauvegardée comme « Définition de Type » ou « Définition de Type Stricte ».

2 9) Considérons le programme suivant :



Avant d'exécuter le programme, la face avant avait l'allure suivante :



On lance maintenant l'exécution du programme et **on ne touche à aucun objet** de la face avant.

Que va-t-il se passer ?

La propriété Val(sgn) (ou valeur signalisation) provoque un événement « Valeur Changée ». En conséquence, la structure événement s'exécutera une fois.

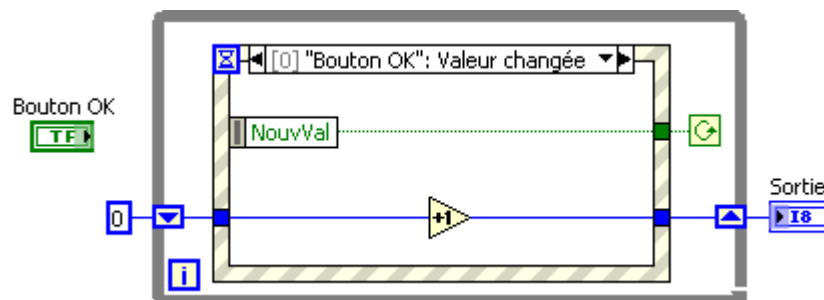
Qu'indiqueront les 3 objets de la face avant ?

Glissière -> restera inchangée

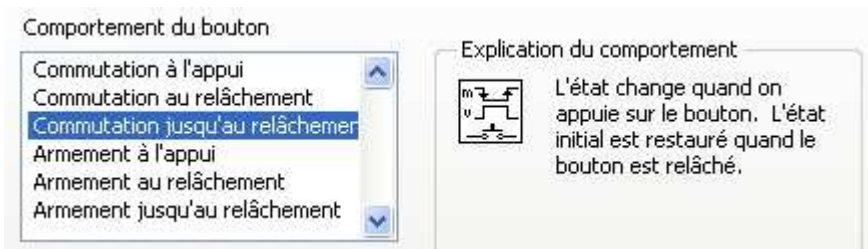
Vumètre -> recopiera la glissière (valeur 6)

Boucle -> valeur 1.

2 10) Considérons le programme suivant :



L'action mécanique du bouton OK est programmée sur commutation jusqu'au relâchement. L'état initial du bouton au repos est False.

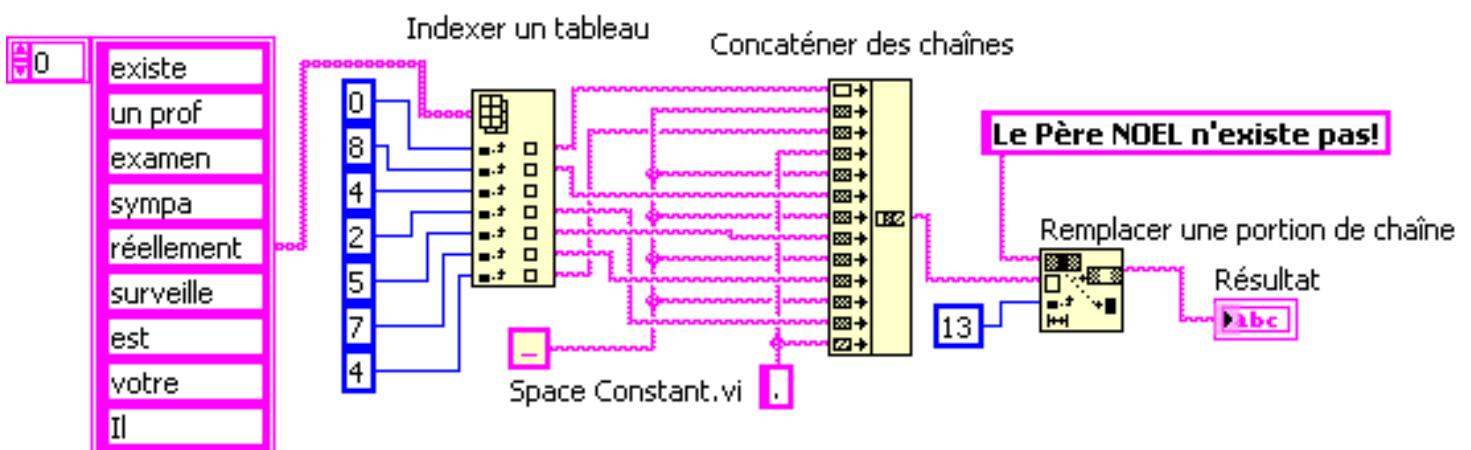


On démarre le programme puis on appuie et on relâche ensuite le bouton. Que va-t-il se passer ?

On notera que le terminal de condition de sortie de boucle est programmé sur « continuer si vrai ».

Dès l'appui sur le bouton OK, l'événement « valeur changée » s'exécute. Le registre à décalage prend alors la valeur 1 et la boucle recommence à attendre un nouvel événement. Au relâchement du bouton OK l'événement « valeur changée » s'exécute à nouveau. Le registre à décalage prend alors la valeur 2. La condition de sortie de boucle est maintenant vérifiée. Le programme s'arrête.

2 11) Considérons le programme suivant :



Donnez la valeur de l'indicateur Résultat en fin d'exécution.

Résultat  
Le Père NOEL existe réellement. Il surveille votre examen.