4. **Peloton d'exécution.** En utilisant la sémantique formelle du langage, exécutez les deux programmes suivants. Faites attention à l'ordre des instructions dans chacun des cas!

```
%% Programme 1
local Res in
   local Arg1 Arg2 in
      Arg1=7          % 1
      Arg2=6          % 2
      Res=Arg1*Arg2   % 3
   end
   {Browse Res}
end
```

```
%% Programme 2
local Res in
   local Arg1 Arg2 in
      Arg1=7          % 1
      Res=Arg1*Arg2   % 3
      Arg2=6          % 2
   end
   {Browse Res}
end
```

L'environnement de départ est {`Browse` → *browse*}. Il mentionne l'identificateur `Browse`, qui est lié à une variable *browse*, que nous supposons définie par le système. Par souci de lisibilité, nous écrivons les variables en italique. La pile sémantique initiale pour le programme 1 est donc

$$\left[\left(\begin{array}{l}\texttt{local Res in}\\ \quad\texttt{local Arg1 Arg2 in}\\ \quad\quad\texttt{Arg1=7}\\ \quad\quad\texttt{Arg2=6}\\ \quad\quad\texttt{Res=Arg1*Arg2}\\ \quad\texttt{end}\\ \quad\texttt{\{Browse Res\}}\\ \texttt{end}\end{array}\right), \{\texttt{Browse} \to \textit{browse}\}\right], (\textit{browse} = (\texttt{proc \{\$ X\} ... end}, \ldots))$$

L'effet de l'instruction `local` est de créer une variable en mémoire, disons *res*, et d'ajouter l'association `Res` → *res* à l'environnement. La pile sémantique devient

$$\left[\left(\begin{array}{l}\texttt{local Arg1 Arg2 in}\\ \quad\texttt{Arg1=7}\\ \quad\texttt{Arg2=6}\\ \quad\texttt{Res=Arg1*Arg2}\\ \texttt{end}\\ \texttt{\{Browse Res\}}\end{array}\right), \left\{\begin{array}{l}\texttt{Browse} \to \textit{browse}\\ \texttt{Res} \to \textit{res}\end{array}\right\}\right], \left(\begin{array}{l}\textit{browse} = (\texttt{proc \{\$ X\} ... end}, \ldots)\\ \textit{res}\end{array}\right)$$

Ensuite, il faut séparer la séquence d'instructions puisqu'on ne peut en exécuter qu'une seule à la fois. La pile sémantique devient

$$\left[\begin{array}{l}\left(\begin{array}{l}\texttt{local Arg1 Arg2 in}\\ \quad\texttt{Arg1=7}\\ \quad\texttt{Arg2=6}\\ \quad\texttt{Res=Arg1*Arg2}\\ \texttt{end}\end{array}, \left\{\begin{array}{l}\texttt{Browse} \to \textit{browse}\\ \texttt{Res} \to \textit{res}\end{array}\right\}\right)\\ \left(\texttt{\{Browse Res\}}, \left\{\begin{array}{l}\texttt{Browse} \to \textit{browse}\\ \texttt{Res} \to \textit{res}\end{array}\right\}\right)\end{array}\right], \left(\begin{array}{l}\textit{browse} = (\texttt{proc \{\$ X\} ... end}, \ldots)\\ \textit{res}\end{array}\right)$$

Continuez l'exécution du programme, en notant bien ce qui se trouve en mémoire. Ensuite, faites de même pour le programme 2. Quel est précisément l'état final de chaque programme? Qu'affichent-ils chacun? Quelles sont les variables créées par chacun? Que deviennent ces variables lorsque les programmes se terminent?

$$\left[\begin{array}{l}\left(\begin{array}{l}\texttt{Arg1} = 7\\ \texttt{Arg2} = 6\\ \texttt{Res} = \texttt{Arg1} * \texttt{Arg2}\end{array}, \left\{\begin{array}{ll}\texttt{Browse} & \to \textit{browse}\\ \texttt{Res} & \to \textit{res}\\ \texttt{Arg1} & \to a_1\\ \texttt{Arg2} & \to a_2\end{array}\right\}\right)\\ \left(\texttt{\{Browse Res\}}, \left\{\begin{array}{l}\texttt{Browse} \to \textit{browse}\\ \texttt{Res} \to \textit{res}\end{array}\right\}\right)\end{array}\right], \left(\begin{array}{l}\textit{browse} = (\texttt{proc \{\$ X\} ... end}, \ldots)\\ \textit{res}, a_1, a_2\end{array}\right)$$

Il faut séparer les 3 instructions séparément avec chacune leur environnement contextuel mais par simplicité on va faire tout d'un coup.

$$\left[\left(\{\text{Browse Res}\}, \left\{ \begin{matrix} \texttt{Browse} \to browse \\ \texttt{Res} \to res \end{matrix} \right\} \right) \right], \left(\begin{matrix} browse = (proc \dots) \\ a_1 = 7, a_2 = 6, res = 42 \end{matrix} \right) \Bigg]$$

On va donc Browse 42

```
%% Programme 2
local Res in
   local Arg1 Arg2 in
      Arg1=7          % 1
      Res=Arg1*Arg2   % 3
      Arg2=6          % 2
   end
   {Browse Res}
end
```

Ici, on va avoir un problème quand on va diviser les 3 instructions car Res va aller chercher la variable a2 en mémoire mais aucune valeur ne lui est associée