Notations: Dans cet énoncé, l'application du lambda-calcul sera notée $(t)\ u$. On notera $\lambda x_1 x_2.t$ pour $\lambda x_1.\lambda x_2.t$, et $(t)\ u_1 \ldots\ u_n$ pour $(\ldots((t)\ u_1)\ldots u_n)$ (précisez si vous adoptez d'autres notations).

Exercice 1:
On va chercher à typer le lambda-terme suivant dans EAL et LAL:

$$t = \lambda w.(w)\ succ\ id\ \underline{0},$$

où $\underline{0}$, $succ$ et id sont respectivement le zéro des entiers de Church, le successeur sur ces entiers et l'identité: $\underline{0} = \lambda fx.x$, $succ = \lambda nfx.(f)\ (n)\ f\ x$.   .

1. On note $N_\alpha = (\alpha \to \alpha) \to (\alpha \to \alpha)$ et
   $W_0 = (N_\alpha \to N_\alpha) \to (N_\alpha \to N_\alpha) \to (N_\alpha \to N_\alpha)$.

   Vérifier que t est typable dans les types simples avec le type $W_0 \to N_\alpha$. Vérifier que t est typable avec le type $W \to N$ dans le système F, où W et N sont les types pour listes binaires et entiers de Church vus en cours (vous pouvez choisir la présentation du typage que vous préférez).

   Que calcule le terme t ?

2. On notera resp. $\mathbf{N}$ et $\mathbf{W}$ les types dans EAL pour les entiers de Church et les listes binaires vus en cours. On notera $\mathbf{N}_\alpha$ et $\mathbf{W_0}$ leur instantiation resp. sur le type de base α et sur $\mathbf{N}_\alpha$.

   Donner dans EAL une dérivation des jugements suivants:

$$\begin{array}{rcll} & \vdash & succ : \mathbf{N}_\alpha \multimap \mathbf{N}_\alpha & (1) \\ x : !(\mathbf{N}_\alpha \multimap \mathbf{N}_\alpha) & \vdash & (x)\ \underline{0} : !\mathbf{N}_\alpha & (2) \\ w : \mathbf{W_0} & \vdash & (w)\ succ\ id\ \underline{0} : !\mathbf{N}_\alpha & (3) \end{array}$$

   Donner dans EAL une dérivation de $t : \mathbf{W} \multimap !\mathbf{N}$, éventuellement en vous servant des dérivations précédentes.

3. On note $\mathbf{N^L}$ et $\mathbf{W^L}$ les types des entiers de Church et des listes binaires dans LAL. Donner une dérivation dans LAL du type $\mathbf{W^L} \multimap \S\mathbf{N^L}$ pour t.

4. Donner la traduction en réseau de preuve LLL de la dérivation de type $\mathbf{W^L} \multimap \S\mathbf{N^L}$ pour t.

Exercice 2:
On va définir dans cet exercice une traduction de LLL dans ILL. Par LLL on désigne ici comme vu en cours le fragment implicatif de la logique linéaire light intuitionniste, et par ILL la logique linéaire intuitionniste.

On considère la traduction $(.)^o$ des formules de LLL vers ILL qui consiste à oublier les $\S$:

$$\alpha^o = \alpha, \quad (\S A)^o = A^o, \quad (!A)^o = !A, \quad (A \multimap B)^o = A^o \multimap B^o$$

1. Montrer que $(.)^o$ définit une traduction des preuves de LLL vers les preuves de ILL.

2. Donner une traduction, aussi notée $(.)^o$, des structures de preuves LLL vers les structures de preuves LL et telle que:

   si R est le réseau associé à une preuve LLL π, alors R^o est le réseau LL associé à la preuve π^o.

3. Montrer que si R est un réseau LLL et $R \longrightarrow R'$ par une étape de réduction, alors $R^o \xrightarrow{\star} R'^o$ (une suite d'étapes de réduction dans les réseaux LL). En déduire que $(.)^o$ est une simulation de LLL vers ILL.