Ejercicio 1

Para cada uno de los siguientes pares de tipos `S` y `T` indicar si están en relación de subtipado, es decir, si `S` `≤` `T` o `T` `≤` `S`, o no están relacionados.

a)

```
S = !int.end
T = !float.end
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-OutS]
!float.end ≤ !int.end
```

```
T ≤ S
```

b)

```
S = ?int.end
T = ?float.end
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-InS]
?int.end ≤ ?float.end
```

```
S ≤ T
```

c)

```
S = !(?int.end).end
T = !(?float.end).end
```

```
inciso b)
---------------------     --------- [S-End]
?int.end ≤ ?float.end     end ≤ end
----------------------------------- [S-OutS]
!(?float.end).end ≤ !(?int.end).end
```

```
T ≤ S
```

d)

```
S = !(!int.end).end
T = !(!float.end).end
```

```
inciso a)
---------------------     --------- [S-End]
!float.end ≤ !int.end     end ≤ end
```

```
----------------------------------- [S-OutS]
!(!int.end).end ≤ !(!float.end).end
```

```
S ≤ T
```

e)

```
S = ?(?int.end).end
T = ?(?float.end).end
```

```
inciso b)
---------------------     --------- [S-End]
?int.end ≤ ?float.end     end ≤ end
----------------------------------- [S-InS]
?(?int.end).end ≤ ?(?float.end).end
```

```
S ≤ T
```

f)

```
S = ?(!int.end).end
T = ?(!float.end).end
```

```
inciso a)
---------------------     --------- [S-End]
!float.end ≤ !int.end     end ≤ end
----------------------------------- [S-InS]
?(!float.end).end ≤ ?(!int.end).end
```

```
T ≤ S
```

g)

```
S = ⊕[l1:!int.end, l2:end]
T = ⊕[l1:!int.end]
```

```
             --------- [S-End]
int ≤ int    end ≤ end
---------------------- [S-OutS]
!int.end ≤ !int.end
--------------------------------------- [S-Choice]
⊕[l1:!int.end, l2:end] ≤ ⊕[l1:!int.end]
```

```
S ≤ T
```

h)

```
S = ⊕[l1:!int.end]
T = ⊕[l1:!float.end]
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-OutS]
!float.end ≤ !int.end
--------------------------------- [S-Choice]
⊕[l1:!float.end] ≤ ⊕[l1:!int.end]
```

```
T ≤ S
```

i)

```
S = ⊕[l1:?int.end]
T = ⊕[l1:?float.end]
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-InS]
?int.end ≤ ?float.end
--------------------------------- [S-Choice]
⊕[l1:?int.end] ≤ ⊕[l1:?float.end]
```

```
S ≤ T
```

j)

```
S = &[l1:!int.end]
T = &[l1:!float.end]
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-OutS]
!float.end ≤ !int.end
--------------------------------- [S-Branch]
&[l1:!float.end] ≤ &[l1:!int.end]
```

```
T ≤ S
```

k)

```
S = &[l1:?int.end]
T = &[l1:?float.end]
```

```
               --------- [S-End]
int ≤ float    end ≤ end
------------------------ [S-InS]
?int.end ≤ ?float.end
--------------------------------- [S-Branch]
&[l1:?int.end] ≤ &[l1:?float.end]
```

```
S ≤ T
```

Ejercicio 2

Asumir que `S = !nat.end` y `T = !float.end` y que `Γ ⊢ −1:int` y `Γ ⊢ 5,0:float`.

Indicar si los siguientes términos están bien tipados.

a)

```
                  X        ---------- [T-Nil]
∅ ⊢ -1:int    int ≤ nat    x+:end ⊢ 0
------------------------------------- [T-Out]
x+:!nat.end ⊢ x+!(-1).0                          x-:?nat.end ⊢ x-?(y:int).0
--------------------------------------------------------------------------- [T-Par]
x+:!nat.end, x-:?nat.end ⊢ x+!(-1).0 | x-?(y:int).0
--------------------------------------------------- [T-Res]
∅ ⊢ (νx:S)(x+!(-1).0 | x-?(y:int).0)
```

No está bien tipado porque no vale `int ≤ nat`.

b)

```
                            ---------- [T-Nil]          X        ----------------- [T-Nil]
∅ ⊢ -1:int    int ≤ float    x+:end ⊢ 0            float ≤ int    x-:end, y:int ⊢ 0
--------------------------------------- [T-Out]    -------------------------------- [T-In]
x+:!float.end ⊢ x+!(-1).0                          x-:?float.end ⊢ x-?(y:int).0
------------------------------------------------------------------------------- [T-Par]
x+:!float.end, x-:?float.end ⊢ x+!(-1).0 | x-?(y:int).0
------------------------------------------------------- [T-Res]
∅ ⊢ (νx:T)(x+!(-1).0 | x-?(y:int).0)
```

No está bien tipado porque no vale `float ≤ int`.

c)

```
                       X        ---------- [T-Nil]
∅ ⊢ 5,0:float    float ≤ nat    x+:end ⊢ 0
------------------------------------------ [T-Out]
x+:!nat.end ⊢ x+!(5,0).0                              x-:?nat.end ⊢ x-?(y:int).0
-------------------------------------------------------------------------------- [T-Par]
x+:!nat.end, x-:?nat.end ⊢ x+!(5,0).0 | x-?(y:int).0
--------------------------------------------------- [T-Res]
∅ ⊢ (νx:S)(x+!(5,0).0 | x-?(y:int).0)
```

No está bien tipado porque no vale `float ≤ nat`.

d)

```
                                 ---------- [T-Nil]          X        ----------------- [T-Nil]
∅ ⊢ 5,0:float    float ≤ float    x+:end ⊢ 0            float ≤ int    x-:end, y:int ⊢ 0
-------------------------------------------- [T-Out]    -------------------------------- [T-In]
x+:!float.end ⊢ x+!(5,0).0                              x-:?float.end ⊢ x-?(y:int).0
------------------------------------------------------------------------------------ [T-Par]
x+:!float.end, x-:?float.end ⊢ x+!(5,0).0 | x-?(y:int).0
```

```
------------------------------------------------------ [T-Res]
∅ ⊢ (νx:T)(x+!(5,0).0 | x-?(y:int).0)
```

No está bien tipado porque no vale `float ≤ int`.

e)

```
∅ ⊢ (νx:S)(x+!(-1).0 | x-?(y:float).0)
```

No está bien tipado por la misma razón que el inciso a).

f)

```
       X
{l1,l2} ⊆ {l1}
------------------------------- [T-Branch]
x+:&[l1:end, l2:end] ⊢ x+▷[l1:0]              x-:⊕[l1:end, l2:end] ⊢ x-◁l2.0
---------------------------------------------------------------------------- [T-Par]
x+:&[l1:end, l2:end], x-:⊕[l1:end, l2:end] ⊢ x+▷[l1:0] | x-◁l2.0
--------------------------------------------------------------- [T-Res]
∅ ⊢ (νx:&[l1:end, l2:end])(x+▷[l1:0] | x-◁l2.0)
```

No está bien tipado porque el proceso de la izquierda no ofrece todas las etiquetas definidas en el canal `x+`.

g)

```
∅ ⊢ (νx:&[l1:end, l2:end])(x+▷[l1:0] | x-◁l1.0)
```

No está bien tipado por la misma razón que el inciso f).

h)

```
                 ---------- [T-Nil]                     ---------- [T-Nil]
{l1} ⊆ {l1,l2}    x+:end ⊢ 0                l1 ∈ {l1}    x-:end ⊢ 0
----------------------------- [T-Branch]    ---------------------- [T-Choice]
x+:&[l1:end] ⊢ x+▷[l1:0, l2:0]              x-:⊕[l1:end] ⊢ x-◁l1.0
------------------------------------------------------------------ [T-Par]
x+:&[l1:end], x-:⊕[l1:end] ⊢ x+▷[l1:0, l2:0] | x-◁l1.0
----------------------------------------------------- [T-Res]
∅ ⊢ (νx:&[l1:end])(x+▷[l1:0, l2:0] | x-◁l1.0)
```

Bien tipado.

i)

```
                 ---------- [T-Nil]          X
{l1} ⊆ {l1,l2}    x+:end ⊢ 0                l2 ∈ {l1}    x-:end ⊢ 0
----------------------------- [T-Branch]    ---------------------- [T-Choice]
x+:&[l1:end] ⊢ x+▷[l1:0, l2:0]              x-:⊕[l1:end] ⊢ x-◁l2.0
------------------------------------------------------------------ [T-Par]
x+:&[l1:end], x-:⊕[l1:end] ⊢ x+▷[l1:0, l2:0] | x-◁l2.0
----------------------------------------------------- [T-Res]
∅ ⊢ (νx:&[l1:end])(x+▷[l1:0, l2:0] | x-◁l2.0)
```

No está bien tipado porque el proceso de la derecha selecciona la etiqueta `l2` que no es una etiqueta disponible en el canal `x-`.

Ejercicio 3

Mostrar si los siguientes pares de tipos infinitos están en relación de subtipado.

Para probar que `S` y `T` están en relación de subtipado: `S ≤ T`, tenemos que encontrar un "type simulation" `R` tal que `(S,T)` ∈ `R`.

a)

```
S = μX.!int.!float.X
T = μX.μY.!int.X
```

`R = {(S,T)}`

```
(S,T) ∈ R
unfold(S) = !int.!float.S
unfold(T) = unfold(μY.!int.T) = !int.T
⇒ (!float.S,T) ∈ R, (int,int) ∈ R
```

`R = {(S,T), (int,int), (!float.S,T)}`

```
(int,int) ∈ R
⇒ int < int
```

```
(!float.S,T) ∈ R
unfold(T) = !int.T
⇒ (S,T) ∈ R, (int,float) ∈ R
```

`R = {(S,T), (int,int), (!float.S,T), (int,float)}`

```
(int,float) ∈ R
⇒ int < float
```

`∴ (S,T) ∈ R ⇒ S ≤ T`

b)

```
S = μX.!(?float.end).!(?int.end).X
T = μX.!(?float.end).X
```

`R = {(T,S)}`

```
(T,S) ∈ R
unfold(T) = !(?float.end).T
unfold(S) = !(?float.end).!(?int.end).S
⇒ (?float.end,?float.end) ∈ R, (T,!(?int.end).S) ∈ R
```

`R = {(T,S), (?float.end,?float.end), (T,!(?int.end).S)}`

```
(?float.end,?float.end) ∈ R
⇒ (float,float) ∈ R, (end,end) ∈ R
```

`R = {(T,S), (?float.end,?float.end), (T,!(?int.end).S), (float,float), (end,end)}`

```
(T,!(?int.end).S) ∈ R
unfold(T) = !(?float.end).T
⇒ (?int.end,?float.end) ∈ R, (T,S) ∈ R
```

`R = {(T,S), (?float.end,?float.end), (T,!(?int.end).S), (float,float), (end,end), (?int.end,?float.end)}`

```
(float,float) ∈ R
⇒ float < float
```

```
(end,end) ∈ R
⇒ end < end
```

```
(?int.end,?float.end) ∈ R
⇒ (int,float) ∈ R, (end,end) ∈ R
```

`R = {(T,S), (?float.end,?float.end), (T,!(?int.end).S), (float,float), (end,end), (?int.end,?float.end), (int,float)}`

```
(int,float) ∈ R
⇒ int < float
```

`∴ (T,S) ∈ R ⇒ T ≤ S`

c)

```
S = μX.!(?float.end).!(?int.end).X
T = μX.!(?int.end).X
```

`R = {(S,T)}`

```
(S,T) ∈ R
unfold(S) = !(?float.end).!(?int.end).S
unfold(T) = !(?int.end).T
⇒ (?int.end,?float.end) ∈ R, (!(?int.end).S,T) ∈ R
```

`R = {(S,T), (?int.end,?float.end), (!(?int.end).S,T)}`

```
(?int.end,?float.end) ∈ R
⇒ (int,float) ∈ R, (end,end) ∈ R
```

`R = {(S,T), (?int.end,?float.end), (!(?int.end).S,T), (int,float), (end,end)}`

```
(!(?int.end).S,T) ∈ R
unfold(T) = !(?int.end).T
⇒ (?int.end,?int.end) ∈ R, (S,T) ∈ R
```

`R = {(S,T), (?int.end,?float.end), (!(?int.end).S,T), (int,float), (end,end), (?int.end,?int.end)}`

```
(int,float) ∈ R
⇒ int < float
```

```
(end,end) ∈ R
⇒ end < end
```

```
(?int.end,?int.end) ∈ R
⇒ (int,int) ∈ R, (end,end) ∈ R
```

```
R = {(S,T), (?int.end,?float.end), (!(?int.end).S,T), (int,float), (end,end), (?int.end,?int.end), (int,int)}
```

```
(int,int) ∈ R
⇒ int < int
```

```
∴ (S,T) ∈ R ⇒ S ≤ T
```

Ejercicio 4

Para los siguientes términos, indicar si están bien tipados. En caso afirmativo, mostrar sus posibles reducciones.

a)

```
⟨
    let s = create () in
    let a = fork (λx. close (send x true)) (fst s) in
    let b = receive (snd s) in
    close (snd b)
⟩

⟶ [r-create]

(νa)⟨
    let s = (c+, c-) in
    let a = fork (λx. close (send x true)) (fst s) in
    let b = receive (snd s) in
    close (snd b)
⟩

⟶ [r-new][r-thread][r-let]

(νa)⟨
    let a = fork (λx. close (send x true)) (fst (c+, c-)) in
    let b = receive (snd (c+, c-)) in
    close (snd b)
⟩

⟶ [r-new][r-fork]

(νa)(⟨
    let a = () in
    let b = receive (snd (c+, c-)) in
    close (snd b)
⟩ || ⟨
    (λx. close (send x true)) (fst (c+, c-))
⟩)

⟶ [r-new][r-par][r-thread][r-let]

(νa)(⟨
    let b = receive (snd (c+, c-)) in
    close (snd b)
```

```
⟩ || ⟨
    (λx. close (send x true)) (fst (c+, c−))
⟩)

⟶ [r−new][r−par][r−thread][r−snd]

(νa)(⟨
    let b = receive c− in
    close (snd b)
⟩ || ⟨
    (λx. close (send x true)) (fst (c+, c−))
⟩)

⟶ [r−new][r−par][r−thread][r−fst]

(νa)(⟨
    let b = receive c− in
    close (snd b)
⟩ || ⟨
    (λx. close (send x true)) c+
⟩)

⟶ [r−new][r−par][r−thread][r−beta]

(νa)(⟨ let b = receive c− in close (snd b) ⟩ || ⟨ close (send c+ true) ⟩)

⟶ [r−new][r−struct][r−comm]

(νa)(⟨ let b = (true, c−) in close (snd b) ⟩ || ⟨ close c+ ⟩)

⟶ [r−new][r−par][r−thread][r−let]

(νa)(⟨ close (snd (true, c−)) ⟩ || ⟨ close c+ ⟩)

⟶ [r−new][r−par][r−thread][r−snd]

(νa)(⟨ close c− ⟩ || ⟨ close c+ ⟩)

⟶ [r−new][r−close]

(νa)(⟨()⟩ || ⟨()⟩) ≡ ⟨()⟩
```