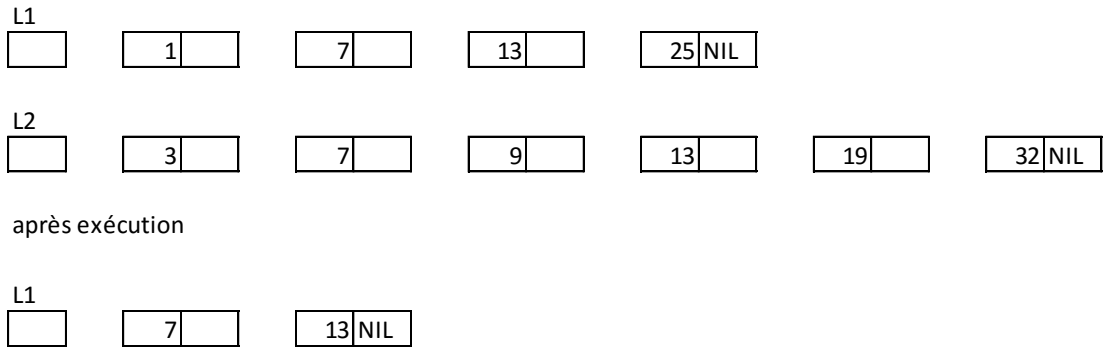
Algorithmique

Devoir maison n°2

Schéma



Hypothèses

- On souhaite conserver L2.
- Les valeurs des deux listes sont triées par ordre croissant, sans répétition.

Principe

```
Pour chaque valeur t1 de L1 Faire

	Recherche de t1 dans L2 à partir de la dernière adresse stockée
	Stockage de l'adresse de la première valeur supérieure ou égale à t1 dans L2

	Si le contenu de ce mot est égal à Nil Alors
		Supprimer les éléments restant de L1
	Sinon

		Si t1 n'est pas dans L2 Alors
			Supprimer t1 dans L1
		Sinon
			Stockage de l'adresse suivant l'adresse de la première valeur
supérieure ou égale à t1 dans L2
		FinSi

	FinSi
FinPour
```

Algorithme

```
Procédure Intersection(e/s: L1, e: L2)

	m(t1) <- cm(L1);
	m(t2) <- cm(L2);
	m(prec) <- L1;

	TantQue cm(cm(t1) + 1) != Nil Faire

		Rech_LchTriée(cm(cm(t1)), cm(t2), valsup, trouvé);

		Si cm(valsup) = Nil Alors

			Sup_Lch_Fin(cm(t1));
			m(cm(prec)) <- Nil;
			m(t1) <- cm(prec) - 1; // Permet de sortir du tant que et
éviter une erreur.

		Sinon

			m(t2) <- cm(valsup);

			Si non(trouvé) Alors
				Sup_Lch(cm(prec));
			Sinon
				m(t2) <- cm(cm(t2) + 1);
				m(prec) <- cm(t1) + 1;
			FinSi

			m(t1) <- cm(cm(t1) + 1);

		FinSi

	FinTantQue

Fin



Procédure Sup_Lch_Fin(e/s: t)

	Si cm(cm(t) + 1) != Nil Alors
		Sup_Lch_Fin(cm(cm(t) + 1));
	FinSi
	Libérer(cm(t), 2);

Fin
```

Lexique

Intersection

L1 : Adresse de la tête de la première liste qui recevra l'intersection.

L2 : Adresse de la tête de la seconde liste.

t1 : Adresse de l'élément courant dans L1.

t2 : Adresse de l'élément courant dans L2, à partir duquel la recherche s'effectue.

prec : Adresse de l'élément précédant l'élément courant, utilisé pour la suppression.

valsup : Adresse dans L2 du premier élément supérieur ou égal à l'élément courant dans L1.

trouvé : Valeur directe, booléenne, contient vrai si l'élément courant est présent dans L2, faux sinon.

Sup_Lch_Fin

t : Adresse de l'élément utilisé pour le parcours de la fin de liste, puis pour sa suppression.

Traces d'exécution

Cas L1 vide

CM(T1)	CM(T2)	CM(PREC)	TROUVE	CM(VALSUP)
A11	A21	L1		

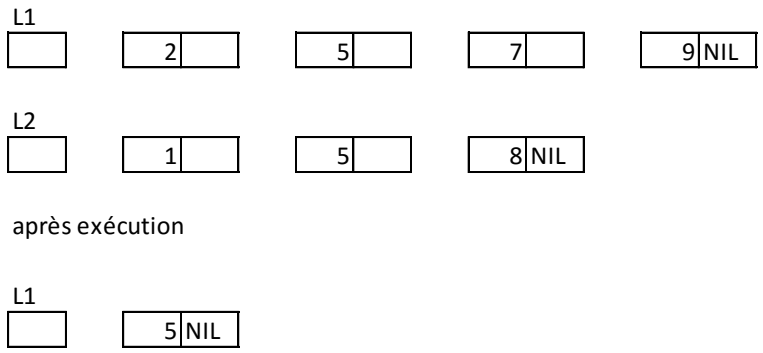
Résultat : L1 est une liste vide.

Cas L2 vide

CM(T1)	CM(T2)	CM(PREC)	TROUVE	CM(VALSUP)
A11	A21	L1		
A12	A22	L1	FAUX	NIL

Résultat : L1 est une liste vide, car elle a entièrement été supprimée.

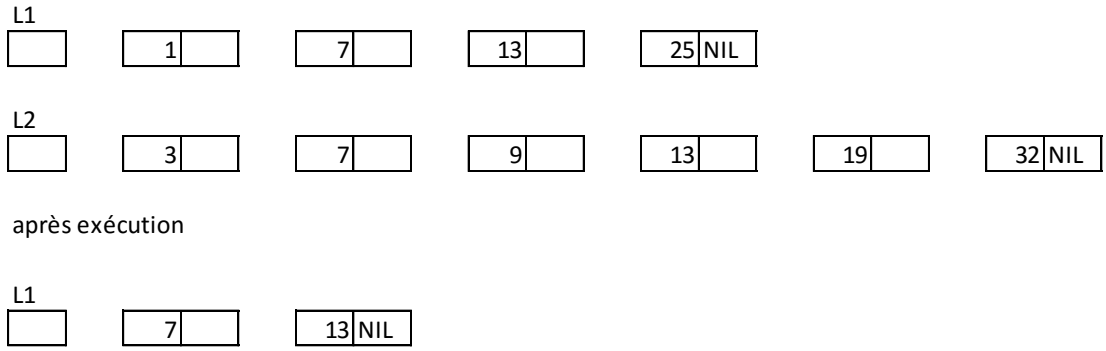
Cas une valeur de L1 est supérieure à toutes celles de L2



CM(T1)	CM(T2)	CM(PREC)	TROUVE	CM(VALSUP)
A11	A21	L1		
A12	A22	L1	FAUX	A22
A13	A23	A12+1	VRAI	A22
A14	A23	A13+1	FAUX	A23
A13	A23	A13+1	FAUX	NIL

Résultat : A la quatrième itération, nous rentrons le cas décrit, la fin de liste est supprimée en partant de l'élément « 9 ».

Cas général



CM(T1)	CM(T2)	CM(PREC)	TROUVE	CM(VALSUP)
A11	A21	L1		
A12	A21	L1	FAUX	A21
A13	A23	A12+1	VRAI	A22
A14	A26	A13+1	VRAI	A26
NIL	A26	A13+1	VRAI	A26

Résultat : L1 est l'intersection des deux listes, les éléments « 7 » et « 13 ».