— *La qualité de la rédaction sera prise en compte. On veillera en particulier à commenter les fonctions écrites ainsi qu'à donner leur signature (y compris pour les éventuelles fonctions auxiliaires).*
— *Les deux exercices sont indépendants. Il est conseillé de passer 75 minutes sur le premier et 45 minutes sur le second.*

Exercice I

On considère l'alphabet $A = \{-1, +1\}$ et on note $A^\star$ l'ensemble des mots (de longueur finie) sur cette alphabet. **On prendra garde au fait que dans ce devoir, les lettres d'un mot de longueur n sont indicées entre 1 et n.**
On note L l'ensemble des mots $w = w_1 w_2 \cdots w_n$ tels que les deux propriétés suivantes soient vérifiées :

— $(P_1) : \sum_{i=1}^{n} w_i = -1$

— (P_2) : pour tout entier k compris (au sens large) entre 1 et $n-1$, $\sum_{i=1}^{k} w_i \geq 0$

Les mots de L s'appellent les mots de Lukasiewicz.
On notera un mot comme une liste d'entiers de type `int list`. On pourra utiliser l'opérateur `@` de concaténation de deux listes.

Quelques propriétés :

1. Donner tous les mots de Lukasiewicz de longueur 1, 2 et 3 puis tous ceux de longueur paire.
2. Écrire une fonction `lukasiewicz : int list -> bool` qui indique si un mot est un mot de Lukasiewicz. Cette fonction renverra une valeur booléenne. La fonction devra avoir une complexité linéaire par rapport à la longueur du mot.
3. (a) Montrer que si $u \in L$ et $v \in L$ alors $(+1).u.v \in L$.
   (b) Réciproquement, montrer que tout mot de Lukasiewicz de longueur supérieure ou égale à 3 admet une décomposition unique sous la forme $(+1).u.v$ où u et v sont des mots de Lukasiewicz.
   (c) Écrire une fonction `decompose : int list -> int list * int list` qui associe ce couple (u, v) à un mot de Lukasiewicz de longueur supérieure ou égale à 3.
4. Écrire une fonction `obtenirLukasiewicz : int -> int list list` qui calcule la liste des mots de Lukasiewicz de taille inférieure ou égale à un entier donné.
   On veillera à ne pas recalculer plusieurs fois les mêmes mots ; on pourra par exemple stocker les mots calculés dans un tableau `t` où `t.(i)` contient les mots de Lukasiewicz de longueur `i`.
   On précisera en français le principe de l'algorithme AVANT de donner le code de la fonction.

Dénombrement :

5. (a) Soit $w = w_1 w_2 \cdots w_n$ un mot de $A^\star$ vérifiant la condition (P_1). Démontrer qu'il existe un unique entier i compris entre 1 et n tel que $w_i w_{i+1} \cdots w_n w_1 w_2 \cdots w_{i-1} \in L$. Ce mot est appelé *conjugué* de w.
   (b) Écrire une fonction `conjugue : int list -> int list` qui associe à tout mot vérifiant (P_1) son conjugué.
6. Déterminer le nombre de mots de Lukasiewicz de longueur $2n+1$.
   On pourra utiliser les résultats précédents ainsi que le corollaire du lemme de Lévi (sans le redémontrer) qui assure que deux mots u et v sur l'alphabet A commutent si et seulement s'il existe $w \in A^\star$ et $p, q \in \mathbb{N}$ tels que $u = w^p$ et $v = w^q$.

Exercice II

Nous nous intéressons dans cet exercice à l'étude de quelques propriétés de la logique propositionnelle tri-valuée. En plus des deux valeurs classiques VRAI ($\top$) et FAUX ($\bot$) que peut prendre une expression, la logique propositionnelle tri-valuée introduit une troisième valeur INDETERMINE (?) .
On considère un ensemble fini de variables propositionnelles noté $\mathcal{V}$ et on note $\mathcal{F}$ l'ensemble des formules construites sur $\mathcal{V}$, sans utiliser les constantes ($\top$, $\bot$ et ?). Pour $A, B \in \mathcal{V}$, les tables de vérités des opérateurs classiques dans cette logique propositionnelle sont les suivantes :

(a) $A \wedge B$

A	B	$A \wedge B$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\top$	?	?
$\bot$	$\top$	$\bot$
$\bot$	$\bot$	$\bot$
$\bot$	?	$\bot$
?	$\top$	?
?	$\bot$	$\bot$
?	?	?

(b) $A \vee B$

A	B	$A \vee B$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\top$	?	$\top$
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\bot$
$\bot$	?	?
?	$\top$	$\top$
?	$\bot$	?
?	?	?

(c) $A \Rightarrow B$

A	B	$A \Rightarrow B$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\top$	?	?
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$
$\bot$	?	$\top$
?	$\top$	$\top$
?	$\bot$	?
?	?	$\top$

(d) $\neg A$

A	$\neg A$
$\top$	$\bot$
$\bot$	$\top$
?	?

Définition 1 : Tri-valuation.
Une tri-valuation est une fonction $f : \mathcal{V} \to \{\top, \bot, ?\}$.

On étend alors de manière usuelle la notion de tri-valuation sur l'ensemble des formules :

Définition 2 : Une tri-valuation sur l'ensemble des formules est une fonction $\widehat{f} : \mathcal{F} \to \{\top, \bot, ?\}$.

Définition 3 : Une tri-valuation $\widehat{f}$ satisfait une formule ϕ si $\widehat{f}(\phi) = \top$. On notera alors $\widehat{f} \vdash_3 \phi$.

Définition 4 : Formule.
Une formule ϕ est :
- une *conséquence* d'un ensemble de formules $\mathcal{X}$ si toute interprétation qui satisfait toutes les formules de $\mathcal{X}$ satisfait ϕ. On notera dans ce cas $\mathcal{X} \Vdash_3 \phi$;
- une *tautologie* si pour toute tri-valuation $\widehat{f}$, $\widehat{f}(\phi) = \top$. On notera dans ce cas $\Vdash_3 \phi$.

1. Montrer que $A \vee \neg A$ n'est pas une tautologie.
2. Proposer alors une tautologie simple dans cette logique.
3. Posons $\top = 1$, $\bot = 0$ et $? = 0,5$. Proposer un calcul simple permettant de trouver la table de vérité de $A \wedge B$ en fonction de A et B. Même question pour $A \vee B$.
4. En logique bi-valuée classique, les propositions $\neg A \vee B$ et $A \Rightarrow B$ sont équivalentes. Qu'en est-il dans le cadre de la logique propositionnelle tri-valuée ?
5. En écrivant les tables de vérité, indiquer si les propositions $\neg B \Rightarrow \neg A$ et $A \Rightarrow B$ sont équivalentes.
6. Donner la table de vérité de la proposition $((A \Rightarrow B) \wedge (\neg A \Rightarrow B)) \Rightarrow B$. Cette proposition est-elle une tautologie ?
7. Un nouvel opérateur d'implication, noté $\rightarrow$, est alors défini, dont la table de vérité est la suivante :

A	B	$A \rightarrow B$
$\top$	$\top$	$\top$
$\top$	$\bot$	$\bot$
$\top$	?	?
$\bot$	$\top$	$\top$
$\bot$	$\bot$	$\top$
$\bot$	?	$\top$
?	$\top$	$\top$
?	$\bot$	?
?	?	?

   (a) La proposition $A \rightarrow A$ est-il une tautologie ?
   (b) Montrer qu'il n'existe aucune tautologie en utilisant uniquement les opérateurs $\wedge$, $\vee$, $\neg$ et $\rightarrow$.