

TD 2 : premiers programmes en Why3

Exercice 1

Soient b un tableau de n entiers, j et k deux indices tels que $0 \leq j < k < n$. Dans ce qui suit, $b[j..k]$ désigne un segment du tableau b . (Cette notation n'existe pas en Why.) Écrire en Why les prédicats exprimant les propriétés suivantes :

1. Tous les éléments de $b[j..k]$ sont nuls.
2. Tous les éléments nuls de $b[0..n-1]$ sont dans $b[j..k]$.
3. Il n'est pas vrai que tous les éléments nuls de $b[0..n-1]$ sont dans $b[j..k]$.
4. Même propriété, mais sans utiliser le symbole de négation.
5. $b[0..n-1]$ contient au moins deux éléments nuls.
6. $b[0..n-1]$ contient deux éléments nuls.
7. $b[0..n-1]$ contient au plus deux éléments nuls.
8. Spécifier la fonction reverse qui reverse l'ordre des éléments d'un tableau d'entiers.

Exercice 2 : factorielle

Spécifier la fonction factorielle en l'axiomatisant.

Écrire à l'aide d'une boucle un algorithme pour la fonction factorielle. Cette fonction aura trois arguments : l'entier pour lequel on calcule factorielle, l'entier contenant le résultat du calcul et un troisième entier utilisé comme temporaire dans la boucle. Dans la boucle de l'algorithme, vous écrirez l'invariant de boucle ainsi qu'un variant.

Exercice 3 : compter dans un tableau

Spécifier et programmer en Why une fonction renvoyant le nombre d'occurrences d'un entier n dans un tableau a d'entiers.

Dans la spécification, vous utiliserez une fonction $nbeq\ a\ v\ i\ j$ du langage logique de Why calculant le nombre d'éléments de $a[i..j]$ valant v . Cette fonction n'est pas à définir; elle sera supposée déjà définie dans une bibliothèque.