Esercizi di Prolog

1. ★☆☆☆☆ Definire un predicato countd/3 tale che countElement(L1, A, C) sia vero se l'elemento A compare C volte in L1.
   Esempio: countElement([a, b, [a, b, c]], a, 1). Restituisce vero

2. ★★☆☆☆ Definire un predicato countd/3 tale che countD(L1, A, C) sia vero se l'elemento A compare C volte in L1, **contando anche le occorrenze nelle sottoliste.**
   Esempio: countD([a, b, [a, b, c]], a, 2). Restituisce vero

3. ★★☆☆☆ Definire un predicato subarray/2 tale che subarray(L1, L2) restituisca vero se L2 è un sottoarray (conta l'ordine e la molteplicità) di L1.
   Esempio: subarray([a, b, c, d], [b, c]). Restituisce vero

4. ★★★☆☆ Definire un predicato subsequence/2 tale che subsequence(L1, L2) restituisca vero se L2 è una sottosequenza di L1, ossia se L2 è costituita da un sottoinsieme di elementi di L1, che rispettano l'ordine originale, ma non sono necessariamente consecutivi. In sostanza è vero se L2 è ottenibile eliminando alcuni (o nessuno) degli elementi di L1.
   Esempio: subsequence([a, b, c, d, e, f], [b, c, e]). Restituisce vero

5. ★★☆☆☆ Definire un predicato flatten/2 tale che flatten(L1, L2) sia vero se L2 è l'elenco degli elementi di L1 e delle sue sottoliste (sostanzialmente "spacchettiamo" le liste **mantenendo l'ordine**).
   Esempio: flatten([a, [b, [c], d], [e, f]], [a, b, c, d, e, f]). Restituisce vero

6. ★★★★☆ Definire un predicato intersection/3 tale che intersection(L1, L2, L3) sia vero se L3 è l'intersezione delle due liste L1 e L2. L'ordine degli elementi non è specificato.
   Esempio: intersection([a, b, c, d, e, f], [e, a, r], [a, e]). Restituisce vero

7. ★★★☆☆ Definire un predicato leafSum/2 tale che leafSum(N, S) sia vero se la somma delle foglie dell'albero con radice N è pari a S (N è nella forma node(Key, Value, LeftChild, RightChild)).
   Esempio: leafSum(node(a, 5, node(b, 3, void, void), node(c, 9, void, void), 12). Restituisce vero

8. ★★★☆☆ Definire un predicato nodeNumber/2 tale che nodeNumber(N, S) sia vero se il numero di nodi dell'albero con radice N è pari a S (N è nella forma node(Key, Value, LeftChild, RightChild)).
   Esempio: nodeNumber(node(a, 5, node(b, 3, void, void), node(c, 9, void, void), 3). Restituisce vero

9. ★★★★☆ Definire un predicato InternalNodeNumber/2 tale che InternalNodeNumber(N, S) sia vero se il numero di nodi **che non sono foglie** dell'albero con radice N è pari a S (N è nella forma node(Key, Value, LeftChild, RightChild)).
   Esempio: nodeNumber(node(a, 5, node(b, 3, void, void), node(c, 9, void, void), 1). Restituisce vero

10. ★★★★★ Definire un predicato lowestCommonAncestor/4 tale che lowestCommonAncestor(A, N1, N2, K) sia vero se il nodo con chiave K è il più basso antenato comune tra N1 e N2 nell'albero che ha radice A (i nodi sono nella forma node(Key, Value, LeftChild, RightChild)). Sostanzialmente state cercando la radice del minimo sottoalbero che contiene sia N1 che N2.
    Esempio: nodeNumber(node(a, 5, node(b, 3, void, void), node(c, 9, void, void), node(b, 3, void, void), node(c, 9, void, void), a). Restituisce vero