



# Prolog exercises

# Hints

---

- ▶ Focus on the solutions, in particular on the features of the result
- ▶ The result is a parameter of the predicate
- ▶ Variables start with a capital letter, atoms with a lower case letter
- ▶ Exploit recursion
  - ▶ Base case(s)
  - ▶ Generic case(s)



# Exercises

---

- ▶ Define a predicate `conta` that counts the occurrences of an element in a list
- ▶ Define a predicate `naturali` that, given a number  $n$ , computes an ordered list with the first  $n$  natural numbers
- ▶ Define a predicate `ribalta` that computes the reverse of a list
- ▶ Define a predicate `inserisci` that inserts a number in an ordered list
- ▶ Given a list of integer, define a predicate `quadrati` that computes a list with their squares
- ▶ Given a list of elements, define a predicate `sonoListe` that computes a list of booleans that tell whether each element is a list

# Exercises

---

- ▶ A tree can be represented as:
  - ▶ [], or
  - ▶ [root, left, right]
  - ▶ Where root is a value and left and right are two trees
- ▶ Define a predicate `creaAlberoBilanciato` that, given a depth, creates a random tree with the given depth
- ▶ Define a predicate `profondita` that computes the depth of the tree
- ▶ Define a predicate `stampaDF` that prints the elements of the tree in depth-first order
- ▶ Define a predicate `contaNodi` that counts the number of nodes in the tree