

# Prolog Exercises

Module of Logics and Artificial Intelligence course

# Exercise 1

Scrivere un programma Prolog che data una lista di liste di interi dia come risultato la lista degli elementi minimi di ogni lista.

Es `minlist([[1,2,3],[6,7],[9,1]], X).`

`true`

`X = [1,6,1]`

## Exercise 2

Si scriva un programma Prolog che data una lista di liste di interi calcoli la somma degli elementi per ogni lista e dia come risultato la somma minima

Es `minSumList([[1,2,3],[6,7],[9,1]], X).`

True

X = 6

## Exercise 3

Scrivere un programma Prolog che calcola la frequenza relativa (numero di occorrenze diviso il numero totale) di una parola in una lista di parole.

Nel caso la lunghezza della lista sia meno di 10 o maggiore di 100, il programma restituisce sempre 0.1 come frequenza relativa.

## Exercise 4

Scrivere un programma Prolog che, date due liste in input, se la loro lunghezza è la stessa, calcola il numero di elementi uguali nella stessa posizione. Se la lunghezza è diversa, verifica se la lista più corta è compresa nell'altra, cioè se gli elementi compaiono nello stesso ordine.

Es. [1,2,3],[9,2,1] il numero di elementi uguali nella stessa posizione è 1

[1,2,3,4,5],[2,3,4] true

[1,2,3,4,5],[1,2,4] false

## Exercise 5

Si scriva un programma Prolog che, data una lista di elementi e un numero intero N, restituisca il prefisso di lunghezza N, cioè la lista dei primi N elementi.  
Successivamente il programma elimina i duplicati dal prefisso.

ES `prd([1,1,2,3],3,O).`  
`O = [1,2]`

## Exercise 6

Si scriva un programma Prolog che, date due liste A e B, restituisca in output due liste C e D tali che C contenga gli elementi di A che appartengono anche a B, mentre D contenga gli elementi di A che non appartengono a B.

## Exercise 7

Si scriva un predicato Prolog `list_to_set` a due argomenti che data una lista di liste come primo argomento leghi il secondo argomento a una lista nella quale sono state eliminate le liste ripetute o le loro permutazioni.



## Exercise 8

Si scriva l'albero di risoluzione SLD completo per il seguente programma Prolog data la query  $p(Y, X)$ .

$p(a,b).$

$p(C,b):- c(C), C \neq b.$

$p(C,D):- d(D),!, c(C).$

$c(a).$

$c(b).$

$d(a).$

## Exercise 9

Si scriva l'albero completo di risoluzione SLD per il seguente programma Prolog data la query  $p(X,Y)$ .

$p(X,Y) \text{ :- } q(X), z(X), r(Y).$

$p(X,Y) \text{ :- } r(X), !, q(Y).$

$p(e,a).$

$q(a).$

$q(X) \text{ :- } s(X).$

$s(x).$

$r(y).$

$z(a).$

## Exercise 10

Si scriva l'albero completo di risoluzione SLD per il seguente programma Prolog data la query  $p(X,Y)$ .

$p(X,Y) \text{ :- } q(X), r(Y).$

$p(a,_) \text{ :- } !.$

$p(e,a).$

$q(X) \text{ :- } s(X).$

$s(x).$

$r(y).$

## Exercise 11

Si scriva l'albero completo di risoluzione SLD per il seguente programma Prolog data la query  $p(X,Y)$ .

$p(X,Y) \text{ :- } q(X), z(Y), r(Y).$

$p(X,Y) \text{ :- } r(X), !, q(Y).$

$p(X,Y) \text{ :- } s(X), q(Y).$

$q(a).$

$q(X) \text{ :- } s(X).$

$s(f).$

$r(y).$

$r(g).$

$z(a).$

## Exercise 12

Si scriva l'albero di risoluzione SLD per il seguente programma Prolog data la query  $u(1,2)$

$u(A,B) :- s(A,B), !, s(B,A).$

$u(\_,\_).$

$s(A,A) :- !.$

$s(A,B) :- s(A), !, s(B).$

$s(\_,\_).$

$s(1).$

$s(2).$