

# Homework 2: PROLOG

## Esercizio 1

Scrivere i seguenti programmi PROLOG:

- `fibonacci(N, F)`, true quando  $F$  è l' $N$ -esimo numero della successione di Fibonacci. La successione di Fibonacci è definita nel modo seguente: i primi due numeri sono 0 e 1; il numero  $n$ -esimo della successione è la somma del  $(n-1)$ -esimo numero e del  $(n-2)$ -esimo numero (ad es. il terzo numero è  $0+1=1$ , il quarto  $1+1=2$ , ecc.).
- `prime(N)`, true quando  $N$  è un numero primo.

## Esercizio 2

Scrivere i seguenti programmi PROLOG:

- `cdot(V, W, X)`, true se  $X$  è il prodotto scalare tra i vettori  $V$  e  $W$ , rappresentati da liste di interi.
- `steep(L)`, true se  $L$  è una lista ripida. Una lista è *ripida* quando ogni elemento è almeno grande quanto la somma di tutti gli elementi che lo precedono.
- `seg(S, L)`: true se  $S$  è un segmento di  $L$ . Un *segmento* di  $L$  è una sottolista di elementi adiacenti di  $L$ . Ad esempio, i segmenti della lista  $[1,2,3]$  sono  $[[1,2,3], [1,2], [2,3], [1], [2], [3], []]$ .
- `isort(X, L, L1)`, true se  $L1$  è una lista ottenuta aggiungendo  $X$  alla lista  $L$  mantenendo l'ordinamento (in ordine crescente) della lista finale. Assumere che la lista  $L$  sia ordinata.

## Esercizio 3

Scrivere il programma PROLOG `dfv(R, N)`, che effettua la **visita depth-first** di un grafo (che può anche **contenere cicli**). Più precisamente, se il grafo ha  $n$  nodi e la visita parte dal nodo  $a$ , le prime  $n$  soluzioni della query `dfv(a, N)` devono ritornare ogni nodo  $N$  da visitare nell'ordine corretto. Per rappresentare il grafo, si usi il predicato `arc(a, b)`, che è true se vi è un arco tra il nodo  $a$  e il nodo  $b$ . Per testare la visita, si può usare il grafo definito nel template fornito.

**Esempio:** nel grafo definito da:

```
arc(a,b).
arc(a,c).
arc(b,d).
arc(b,e).
arc(c,b).
arc(e,c).
```

le prime 5 soluzioni della query devono  $dfv(a, N)$  essere:

$N = a$  ;

$N = b$  ;

$N = d$  ;

$N = e$  ;

$N = c$  ;

...

## Regole di consegna

- Va consegnato solo il file “studente.pl”, **rinominato in <matricola>.pl (ad es. “123456.pl”)** contenente, oltre alla knowledge base fornita, i predicati degli esercizi, con scadenza **5 giorni prima della data dell’esame** che si intende sostenere.
- Gli homework sono validi **solo per gli appelli di esame di Giugno e Luglio e sostituiscono la prova orale.**

## Valutazione

Ogni homework è valutato da 0 a 30 punti. Il punteggio finale per questa parte del corso verrà calcolato come punteggio  $(voto\_hw1 + voto\_hw2 + 2 * voto\_scritto) / 4$ .