

### ESERCIZIO 1 [15 punti]

Scrivere un predicato Prolog **max\_seqs(L)**, dove L è una lista di interi che stampi la più estesa sequenza crescente e la più estesa sequenza decrescente.

```
?- max_seqs([1,2,1,2,3,4,2,1,2,1]).
max sequenza crescente [1, 2, 3, 4]
max sequenza decrescente [4, 2, 1]
true
?- max_seqs([1,2,1,3,1,3,4,5,6,3,2,1,2,3,2,1]).
max sequenza crescente [1, 3, 4, 5, 6]
max sequenza decrescente [6, 3, 2, 1]
true
```

### ESERCIZIO 2 [4 punti]

Scrivere un predicato Prolog **cartprod(X)**, che legge dall'input due qualunque liste e, dopo aver controllato che effettivamente siano liste ben formate di qualcosa, restituisca in X il loro prodotto cartesiano.

```
?- cartprod(X).
|: [a,b,c].
|: [1,2].
X = [ (a, 1), (a, 2), (b, 1), (b, 2), (c, 1), (c, 2)].
```

### ESERCIZIO 3 [11 punti]

Definire un predicato Prolog **take(List,N,Result)** che ha successo se e solo se **Result** è la lista costituita dai primi **N** elementi di **List** (nello stesso ordine). Se **List** ha meno di **N** elementi o se **N** è zero o negativo, il goal **take(List,N,Result)** fallisce segnalando il tipo di errore. Definire inoltre un predicato **drop(List,N,Result)** che ha successo se e solo se **Result** è la lista costituita dagli elementi di **List** (nello stesso ordine) che rimangono quando sono stati eliminati i primi **N** elementi. Se **List** ha meno di **N** elementi o se **N** è zero o negativo, il goal **drop(List,N,Result)** fallisce segnalando il tipo di errore.

```
?- take([a,b,c,d,e,f],4,Result).
Result = [a, b, c, d].
?- take([a,b,c,d,e,f],-4,Result).
ERRORE: N non positivo
false.
?- take([a,b,c,d,e,f],8,Result).
ERRORE: La lista ha meno di 8 elementi!!
false.

?- drop([a,b,c,d,e,f],4,Result).
Result = [e, f].
?- drop([a,b,c,d,e,f],-4,Result).
ERRORE: N non positivo
false.
?- drop([a,b,c,d,e,f],8,Result).
ERRORE: La lista ha meno di 8 elementi!!
false.
```