

Esercizio 1

Si scriva un predicato Prolog che data una lista ed un elemento El appartenente alla lista, restituisca in uscita l'elemento successivo ad El nella lista. Nel caso in cui El sia l'ultimo elemento il predicato dovrà fallire.

Esempio:

```
?- consec(3, [1,7,3,9,11],X).  
yes X=9
```

Esercizio 2

Avendo definito

```
pari(X) :- 0 is X mod 2.
```

definire il predicato `split(+L,-P,-D)` tale che se L è una lista di interi, P sarà la lista contenente tutti gli elementi pari di L e D tutti quelli dispari (nello stesso ordine in cui occorrono in L).

Esempio:

```
?- split([1,2,3,7,8],P,D).  
yes, D=[1,3,7], P=[2,8]
```

Esercizio 3

Data una lista L1 e un numero intero N, scrivere un predicato Prolog `domanda1(+L1,+N,-L2)` che restituisca in L2 la lista degli elementi di L1 che sono liste contenenti solo due valori interi positivi fra 1 e 9 la cui somma valga N.

Esempi:

```
?- domanda1([[3,1],5,[2,1,1],[3],[1,1,1],a,[2,2]],4,L2).  
yes, L2 = [[3,1],[2,2]]
```

Esercizio 4

Si definisca un predicato in PROLOG chiamato `averStud` che applicato a un numero di matricola di uno studente `Matr` e a una lista di esami `LE` dia come risultato la media `AV` dei suoi voti. Ogni esame sia rappresentato da un termine della lista `LE` della forma `esame(Mat,Esame,Voto)`. Si definisca prima la versione ricorsiva e poi quella ricorsiva-tail.

Esempio:

```
?- averStud(s1,[esame(s2,f1,30),  
esame(s1,f1,27),esame(s3,f1,25),  
esame(s1,f2,30)],AV).  
yes, AV = 28.5
```