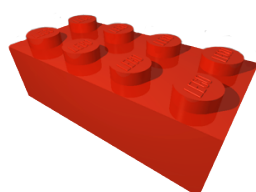


CORSO DI LINGUAGGI DI PROGRAMMAZIONE

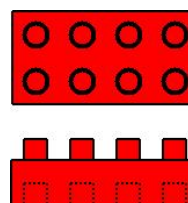
Progetto A.A. 2007/2008

C. Braghin e A. Ceselli

Le costruzioni LEGO (contrazione dell'espressione danese di "giocare bene") sono probabilmente uno dei giocattoli più famosi al mondo. La ditta danese cominciò a produrre giocattoli di legno nel 1932, ma divenne famosa per i mattoncini colorati ad incastro (vedi figura 1(a)), prodotti a partire dal 1949.



(a) Mattoncino 2x4



(b) Schema

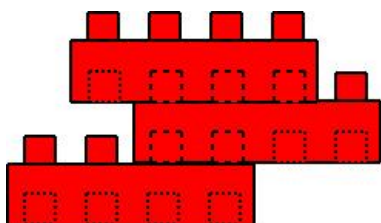
Figura 1: Mattoncini LEGO.

Si tratta di mattoncini a forma rettangolare di diversa dimensione, composti da un certo numero di moduli ad incastro disposti in righe e colonne. Ad esempio, il mattoncino in figura 1(a), schematizzato in figura 1(b), è composto da otto moduli ad incastro, disposti su due righe e quattro colonne (2x4 per essere brevi).

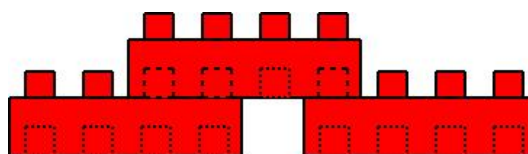
Le costruzioni LEGO sono considerate uno dei giocattoli più versatili e più adatti a stimolare la creatività dei bambini. Ed infatti, a scopi pubblicitari e divulgativi, l'azienda afferma dal 1974 che il numero di modi diversi per combinare sei mattoncini dello stesso colore e di dimensione 2x4 sia 102, 981, 500.

Sorprendentemente, questo numero è stato dimostrato essere sbagliato solo pochi anni fa.

Cosa fare: Implementare un algoritmo che, ricevendo in input un numero k , sia in grado di fornire un elenco di tutti i modi diversi in cui possono essere *impilati* k mattoncini 2x4. Per *impilati* si intende che tutti i mattoncini devono essere incastrati tra di loro, e la parte inferiore di ogni mattoncino (ad esclusione del primo) deve essere incastrata alla parte superiore di un solo altro mattoncino (vedi figura 2(a)). Ne segue che nessuna coppia di mattoncini può essere appaiata (vedi figura 2(b)).



(a) Configurazione in pila



(b) Configurazione non in pila

Figura 2: Incastri tra i mattoncini.

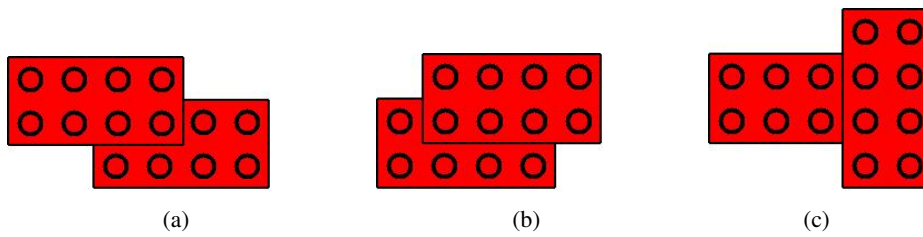


Figura 3: Due mattoncini impilati.

Ovviamente, è necessario tenere in considerazione che due mattoncini possono essere impilati in molti modi (vedi figura 3).

Possibili varianti:

- Modificare l’algoritmo in modo che sia in grado di *contare* il numero di configurazioni prodotte.
- Modificare l’algoritmo in modo che consideri *un solo rappresentante* per ogni insieme di pile dalla forma *simmetrica* (vedi figura 4 e 5).
- Modificare l’algoritmo in modo che sia in grado di elencare tutte le configurazioni in cui possono essere incastrati i k mattoncini, e non solamente le pile.

Linee guida: L’algoritmo deve essere implementato in tre versioni, utilizzando tre linguaggi diversi: Prolog, Scheme ed un linguaggio a scelta. L’implementazione nel linguaggio a scelta deve rispettare il paradigma di programmazione imperativo. Per l’implementazione con paradigma imperativo verranno premiate scelte di linguaggi di programmazione poco diffusi. Non è importante l’*efficienza*, ma lo *stile* del codice prodotto. Consegnare il codice sorgente ed una relazione (preferibilmente in formato pdf). Nella relazione non è particolarmente importante descrivere l’algoritmo utilizzato quanto riportare le proprie osservazioni e critiche, evidenziando *vantaggi e svantaggi* dell’utilizzo di ciascun linguaggio e *differenze* nelle tre implementazioni.

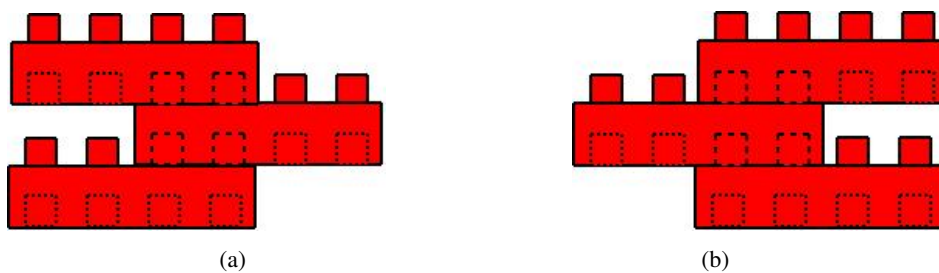


Figura 4: Esempio di pile simmetriche (1).

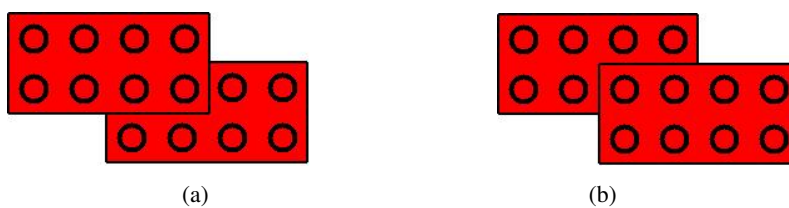


Figura 5: Esempio di pile simmetriche (2).