

Strutture dati

1

Scrivere un programma che gestisca la collisione in una dimensione tra due blocchi caratterizzati da due proprietà:

- massa
- velocità

La formula per calcolare la velocità finale è la seguente:

$$V_1 = -\frac{m_2 \cdot v_2}{m_1}$$
$$V_2 = -\frac{m_1 \cdot v_1}{m_2}$$

Dove m_1 e m_2 sono le masse dei due blocchi, v_1 e v_2 le velocità iniziali e V_1 e V_2 le velocità finali.

Basta che il programma gestisca una sola collisione e stampi le velocità finali.

2

Scrivere un programma per permettere la ricerca in un albero genealogico tramite il nome di una persona.

Aiuto: Definire una persona con i campi

- nome
- madre
- padre

dove **madre** e **padre** sono a loro volta struct di tipo **Persona**.

Aiuto: inizializzare le persone che chiudono l'albero (es. i nonni) con NULL, in modo da poter controllarne la presenza ed evitare seg. fault.

3

Scrivere un programma che permetta la gestione di una serie di macchine tramite `LinkedList`, definita nel seguente modo:

```
1 struct LinkedList {  
2   // attributi utili  
3   // (...)  
4   // puntatore all'item successivo nella lista  
5   LinkedList * next;  
6 };
```

3.1

Scrivere una funzione che permetta di stampare tutti gli item all'interno della lista.

3.2

Scrivere una funzione che ricerchi un determinato item all'interno della lista e lo rimuova.

N.B: non è necessario de-allocare l'oggetto, basta manipolare il campo `next`

Usare la funzione creata in 3.1 per controllare che l'oggetto sia effettivamente stato rimosso