

1. (**4 punti**) Scrivere una funzione C *iterativa* contenuto `It` che ricevendo come parametri di input due vettori di interi a e b e le relative lunghezze, restituisca:
 - (a) come valore di ritorno 1 se a è contenuto in b , -1 se b è contenuto in a e 0 altrimenti.
 - (b) nel caso la risposta sia 1 la funzione deve ritornare (in due parametri passati per indirizzo) anche gli indici di b dove occorrono il primo e l'ultimo elemento di a .
 - (c) nel caso la risposta sia -1 la funzione deve ritornare (in due parametri passati per indirizzo) anche gli indici di a dove occorrono il primo e l'ultimo elemento di b .
2. (**4 punti**) Scrivere una funzione C *ricorsiva* contenuto `Rec` che risolve lo stesso problema.

Note: Sia nel caso iterativo che in quello ricorsivo potete scrivere funzioni ausiliarie.

Nei punti 1b e 1c, la risposta non è unica (come potete osservare nell'esempio, ci sono altri 1 a sinistra del primo 1 evidenziato in grassetto e a destra dell'ultimo 1 in grassetto).

Esercizio 3 (8 punti) Si progetti un algoritmo che, dato un insieme S di n numeri reali ed un altro numero reale x , in tempo $O(n \log n)$ determini se esistono due elementi di S la cui somma sia esattamente x .

Dell'algoritmo:

1. (**6 punti**) Si scriva una breve descrizione a parole e lo pseudocodice;
2. (**2 punti**) Si valuti la complessità computazionale;

Esercizio 4 (6 punti) Si rappresenti l'albero delle decisioni relativo all'algoritmo di Bubble Sort sul vettore $A = [a_1, a_2, a_3]$.