
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 30 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `count_triplets_sum_k` con tipo di ritorno `int` che prende come parametri un array di interi `arr`, la sua dimensione `n`, e un intero `k`. La funzione deve ritornare il numero di triplete distinte di indici (i, j, l) tali che $i < j < l$ e

$$\text{arr}[i] + \text{arr}[j] + \text{arr}[l] = k.$$

Per esempio, se `arr = [1, 2, 3, 4, 5]` e `k = 9`, la funzione deve ritornare 2 (le triplete valide sono $(0, 2, 4)$ con $1 + 3 + 5 = 9$ e $(1, 2, 3)$ con $2 + 3 + 4 = 9$).

Esercizio 2 (9pt).

- Scrivere una funzione `addSpaceAfterDots` che prende come parametro una stringa C-style `str` e ritorna una nuova stringa in cui, dopo ogni punto '.', è aggiunto uno spazio, a meno che il punto non sia già seguito da uno spazio.
- Scrivere una funzione `main` che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione `addSpaceAfterDots` passando come primo parametro la stringa letta e infine stampa il risultato ritornato dalla chiamata alla funzione `addSpaceAfterDots`. Nel caso in cui la lunghezza della stringa inserita sia maggiore di 100 caratteri, la funzione `main` deve ritornare -1.

Esercizio 3 (15pt). Scrivere una funzione `unionLists` che, prese come parametri due liste semplicemente concatenate `lst1` e `lst2` i cui elementi hanno campo informazione di tipo `int`, ritorna **una nuova lista** e senza modificare `lst1` e `lst2` contenente tutti gli elementi presenti in `lst1` o in `lst2` (l'unione), senza duplicati.

Ad esempio, se `lst1 = 1 → 2 → 3 → 4` e `lst2 = 2 → 4 → 6`, la funzione deve ritornare una lista `1 → 2 → 3 → 4 → 6`.

Gestire in modo opportuno il caso in cui una o entrambe le liste siano vuote. Si definisca inoltre la struttura che modella un nodo di una lista semplicemente concatenata con campo informazione di tipo `int`.