

---

## FONDAMENTI DI PROGRAMMAZIONE A

---

*Tempo a disposizione: 1 ora 30 minuti*

Nome ..... Cognome ..... Matricola .....

**Esercizio 1 (6pt).** Scrivere una funzione `count_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 coppie di indici  $(i, j)$  tali che  $i < j$  e  $\text{arr}[i] + \text{arr}[j] = k$ . Per esempio, se `arr = [1, 5, 7, -1, 5]` e `k = 6`, la funzione dovrà ritornare 3 (coppie valide:  $(0, 1), (2, 3), (0, 4)$ ).

**Esercizio 2 (9pt).**

- ▶ Scrivere una funzione `removeDigits` che prende come parametro una stringa C-style `str` e ritorna una nuova stringa in cui tutti i caratteri numerici (da 0 a 9) sono stati rimossi.
- ▶ Scrivere una funzione `main` che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione `removeDigits` passando come primo parametro la stringa letta e infine stampa il risultato ritornato dalla chiamata alla funzione `removeDigits`. 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`.