
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 40 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `count` con tipo di ritorno `int` che prende come parametri un array di interi `arr` e la sua dimensione `n`. La funzione deve ritornare il numero di coppie di indici (i, j) tali che $i < j$ e `arr[i] = arr[j]`. Per esempio, se `arr = [1, 2, 1, 2, 1]` la funzione dovrà ritornare 4 (inversioni: (0, 2), (0, 4), (2, 4), (1, 3)).

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 `difference` che, prese come parametri due liste semplicemente concatenate `lst1` e `lst2` i cui elementi hanno campo informazione di tipo `int`, ritorna una nuova lista contenente tutti e soli gli elementi presenti in `lst1` che non compaiono in `lst2`, mantenendo l'ordine crescente. Ad esempio, se `lst1 = 1 → 2 → 3 → 4 → 5` e `lst2 = 2 → 4 → 6`, la funzione deve ritornare la lista `lst = 1 → 3 → 5`. 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`.