
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 40 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `countInversions` 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 inversioni in `arr`, dove un'inversione è una coppia di indici (i, j) tali che $i < j$ e $\text{arr}[i] > \text{arr}[j]$. Per esempio, se `arr = [2, 4, 1, 3, 5]` la funzione dovrà ritornare 3 (inversioni: $(0, 2), (1, 2), (1, 3)$).

Esercizio 2 (9pt).

- ▶ Scrivere una funzione `remove` che, presi come parametri due stringhe C-style `str1` e `str2` ritorna una nuova stringa corrispondente a `str1` senza i caratteri che si trovano in `str2`. Per esempio, se `str1 = "abcdef"` e `str2 = "ac"`, la funzione dovrà ritornare la stringa `"bdef"`.
- ▶ Scrivere una funzione `main` che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione `remove` passando come primo parametro la stringa letta e come secondo parametro la stringa `"ac"` e infine stampa il risultato ritornato dalla chiamata alla funzione `remove`. 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 `merge` che, prese come parametri due liste semplicemente concatenate `lst1` e `lst2` i cui elementi hanno campo informazione di tipo `int`, entrambe ordinate in modo crescente, ritorna una nuova lista che contiene tutti gli elementi di `lst1` e `lst2` mantenendo l'ordine crescente. Ad esempio, se `lst1 = 2 → 4 → 6` e `lst2 = 1 → 3 → 5`, la funzione deve ritornare la lista `lst = 1 → 2 → 3 → 4 → 5 → 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`.