
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 30 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `swap` con tipo di ritorno `void` che prende come parametri due array di interi `a` e `b`, della stessa dimensione `n` presa anch'essa come parametro. La funzione deve scambiare gli elementi di `a` e `b` se questi hanno lo stesso indice e l'elemento di `a` è maggiore di quello di `b`. Per esempio, se `a = [1, 6, 2, 2, 4]` e `b = [3, 4, 5, 4, 1]`, dopo la chiamata di funzione `swap(a, b, 5)`, gli array saranno `a = [1, 4, 2, 2, 1]` e `B = [3, 6, 5, 4, 4]`.

Esercizio 2 (9pt).

- ▶ Scrivere una funzione `char* remove_double(char* str)` che prende come parametro una stringa C-style `str` e ritorna una nuova stringa in cui ogni carattere raddoppiato è ridotto a una singola occorrenza. Ad esempio, se `str = "aabbccc"`, la funzione ritorna la stringa `"abc"`.
- ▶ Scrivere una funzione `main` che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione `remove_double` passando come parametro la stringa letta, ed infine stampa il risultato ritornato dalla chiamata alla funzione `remove_double`. Nel caso in cui l'utente provi ad inserire una stringa di lunghezza maggiore di 100, la funzione `main` deve ritornare -1.

Esercizio 3 (15pt). Scrivere una funzione `remove_duplicates` che prende come parametro una lista semplicemente concatenata `lst` i cui elementi hanno campo informazione di tipo `int` e ritorna una nuova lista senza elementi duplicati. Ad esempio, se `lst = 1 → 2 → 2 → 3 → 3`, la funzione ritornerà la lista `1 → 2 → 3`. Gestire in modo opportuno il caso in cui la lista è vuota. Si scriva inoltre la struttura che modella un nodo di una lista semplicemente concatenata con campo informazione di tipo `int`.