
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 45 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 **replace** che, presi come parametri una stringa C-style **str**, un carattere **target** e un carattere **replacement**, ritorna una nuova stringa corrispondente a **str** in cui tutte le occorrenze del carattere **target** sono sostituite dal carattere **replacement**.
- Scrivere una funzione **main** che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione **replace** passando come parametri la stringa letta, il carattere 'a' e il carattere 'e', ed infine stampa il risultato ritornato dalla chiamata alla funzione **replace**. Nel caso in cui la lunghezza della stringa letta sia maggiore di 100 caratteri, la funzione **main** deve ritornare -1.

Esercizio 3 (15pt). Scrivere una funzione di nome **rotate** che, preso come parametro una lista semplicemente concatenata **lst** i cui elementi hanno campo informazione di tipo **int**, esegua una "rotazione in avanti" della lista: la rotazione dovrà spostare l'ultimo elemento della lista nella posizione in testa. Ad esempio, se **lst** = 1 → 2 → 3 → 4 → 5, dopo la chiamata a funzione **rotate(lst)**, la lista sarà **lst** = 5 → 1 → 2 → 3 → 4. Gestire in modo opportuno il caso in cui **lst** sia vuota. Si scriva inoltre la struttura che modella un nodo di una lista semplicemente concatenata con campo informazione di tipo **int**.