
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 45 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `void shift_left(int arr[], int n, int k)` che prende come parametro un array di interi `arr`, la sua dimensione `n` e un intero `k`, e sposta tutti gli elementi dell'array a sinistra di `k` posizioni, riempiendo con zeri le posizioni vacanti. Per esempio, se `arr = [1, 2, 3, 4, 5]` e `k = 2`, dopo la chiamata `shift_left(arr, 5, 2)`, l'array sarà `arr = [3, 4, 5, 0, 0]`.

Esercizio 2 (9pt).

- Scrivere una funzione `interleave` che, prese come parametri due stringhe C-style `str1` e `str2` di lunghezza uguale, ritorna una nuova stringa in cui i caratteri di `str1` e `str2` sono alternati. Ad esempio, se `str1 = "abcd"` e `str2 = "1234"`, la funzione ritorna la stringa `"a1b2c3d4"`.
- Scrivere una funzione `main` che legge da tastiera due stringhe di massimo 100 caratteri, chiama la funzione `interleave` passando come parametri le stringhe lette, ed infine stampa il risultato ritornato dalla chiamata alla funzione `interleave`. Nel caso in cui la lunghezza delle stringhe lette sia diversa o l'utente provi ad inserire stringhe di lunghezza maggiore di 100, la funzione `main` deve ritornare -1.

Esercizio 3 (15pt). Scrivere una funzione di nome `rotate_back` che, preso come parametro una lista semplicemente concatenata `lst` i cui elementi hanno campo informazione di tipo `int`, esegua una "rotazione all'indietro" della lista: la rotazione dovrà spostare il primo elemento della lista nella posizione in coda. Ad esempio, se `lst = 1 → 2 → 3 → 4 → 5`, dopo la chiamata a funzione `rotate(lst)`, la lista sarà `lst = → 2 → 3 → 4 → 5 → 1`. 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`.