
FONDAMENTI DI PROGRAMMAZIONE A

Tempo a disposizione: 1 ora 40 minuti

Nome Cognome Matricola

Esercizio 1 (6pt). Scrivere una funzione `int find_longest_word_length(char* sentence)` che prende come parametro una stringa C-style `sentence` e ritorna la lunghezza della parola più lunga in `sentence`. Ad esempio, se `sentence = "The quick brown fox jumps over the lazy dog"`, la funzione ritornerà 5.

Esercizio 2 (9pt).

- Scrivere una funzione `char* replace_spaces(char* str)` che prende come parametro una stringa C-style `str` e ritorna una nuova stringa in cui tutti gli spazi sono sostituiti con il carattere underscore '_'. Ad esempio, se `str = "Hello World"`, la funzione ritorna la stringa `"Hello_World"`.
- Scrivere una funzione `main` che legge da tastiera una stringa di massimo 100 caratteri, chiama la funzione `replace_spaces` passando come parametro la stringa letta, ed infine stampa il risultato ritornato dalla chiamata alla funzione `replace_spaces`. 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 di nome `sum_lists` che, prese come parametri due liste semplicemente concatenate `lst1` e `lst2` i cui elementi hanno campo informazione di tipo `int`, ritorna una nuova lista i cui elementi sono la somma degli elementi corrispondenti di `lst1` e `lst2`. Ad esempio, se `lst1 = 1 → 2 → 3` e `lst2 = 4 → 5 → 6`, la chiamata a funzione `sum_lists(lst1, lst2)` ritornerà la lista `5 → 7 → 9`. Se le liste hanno lunghezze diverse, gli elementi extra della lista più lunga devono essere copiati nella lista risultante. Gestire in modo opportuno il caso in cui una delle liste sia vuota. Si scriva inoltre la struttura che modella un nodo di una lista semplicemente concatenata con campo informazione di tipo `int`.