

Laboratorio di programmazione strutturata (STM) - A.A. 2017-2018 -

Primo appello - giugno 2018

ESERCIZI

Tempo a disposizione: 2 ore e mezza. Ogni esercizio vale 10 punti. Per superare l'esame bisogna raggiungere un minimo di 6 punti su ogni esercizio.

Esercizio 1

Si progetti una funzione in C che, data una lista P di punti del piano cartesiano, restituisce la sottolista di P contenente solo i punti che si trovano all'interno del quadrato più piccolo con centro in $(0,0)$ contenente il primo punto della lista P .

1. Si definisca il tipo nodo lista adatto per la lista P .
2. Si implementi in C una funzione denominata `InQuadrato`, che prende in input la lista di punti P e restituisce la lista P modificata come spiegato in precedenza. La funzione non deve creare nuovi nodi ma deve utilizzare quelli della lista di input ed inoltre deve essere preservato l'ordinamento dei nodi rispetto alla lista originale.

Suggerimento: si scorra la lista P e, dopo aver individuato il lato del quadrato, si cancellino tutti i nodi che non stanno all'interno del quadrato.

Esercizio 2 Data una matrice quadrata di dimensione $n \times n$ a valori in un insieme $\{0, 1, \dots, k\}$, il *carico* di una posizione (i, j) , indicato con $c(i, j)$, è dato dalla differenza tra la somma degli elementi della riga i e la somma degli elementi della colonna j . Ad esempio nella seguente matrice:

1	2	1	1
0	0	0	1
1	1	2	0
2	0	0	0

$$c(0,0) = 1, c(1,0) = -3 \text{ e } c(3,3) = 0.$$

1. Scrivere una funzione `caricoNullo` che prende una matrice e la sua dimensione n e restituisce il numero di elementi di carico nullo. L'algoritmo deve utilizzare memoria aggiuntiva costante. Discutere la complessità di tempo dell'algoritmo utilizzato.
2. Mostrare un'altra funzione `caricoNullo2` con complessità di tempo inferiore che però utilizza memoria aggiuntiva $O(n)$.

Esercizio 3

Sia `char cognomeNome[10]`; un vettore inizializzato con le prime 10 lettere del vostro cognome seguite da quelle del vostro nome. Simulate l'algoritmo *selection sort* sul vettore `cognomeNome` mostrando il contenuto del vettore nei vari passi (non occorre eseguire l'algoritmo fino a completo ordinamento del vettore).