

Programmazione I

Prova di programmazione – 22 febbraio 2017 – 2 ore

Partendo dal frammento di codice fornito, realizzare un programma per la gestione dei seguenti quadrilateri convessi: quadrati, rettangoli, trapezi. In particolare, il programma permette di memorizzare una sequenza di tali quadrilateri, e, per ciascun quadrilatero, memorizza la lunghezza di ciascun lato. Per semplicità, le lunghezze devono essere numeri naturali. All'avvio del programma, la sequenza è vuota ed ha lunghezza massima zero (ossia non si può inserire alcun quadrilatero). Il programma deve fornire le seguenti funzionalità.

1. **modifica_lunghezza_massima(N)** Fissa ad **N** il numero massimo di quadrilateri memorizzabili nella sequenza. Il valore di **N** è noto **solo quando questa funzionalità è invocata** (e **non c'è un limite superiore** a tale valore). L'eventuale precedente contenuto della sequenza è perso.
2. **inserisci_quadrilatero()** Inserisce un nuovo quadrilatero in fondo alla sequenza, leggendo da standard input il tipo di quadrilatero e le lunghezze dei lati. Non controlla in alcun modo l'ordine con cui sono inserite le lunghezze dei lati. Tale ordine non ha quindi alcuna importanza. Non controlla neanche la congruenza di tali lunghezze, a parte il fatto che non siano negative o nulle. Per esempio, accetterebbe un quadrato con lati di lunghezza 10, 1, 1, 2.
3. **stampa_sequenza** Stampa il contenuto della sequenza, riportando, per ciascun quadrilatero, il tipo del quadrilatero e le lunghezze dei lati. Quindi potrebbe stampare, ad esempio, per una sequenza contenente tre quadrilateri,
TRAPEZIO 10 10 2 5
QUADRATO 4 4 4 4
RETTANGOLO 5 6 6 5
4. **salva_stato** Salva il contenuto della sequenza in un file di testo dal nome definito a tempo di scrittura del programma.
5. **carica_stato** Carica il contenuto della sequenza dal file. Il contenuto precedente è perso.
6. **controlla_quadrilatero(idx)** Dato il quadrilatero di indice **idx** nella sequenza, ritorna *vero* solo se le lunghezze dei lati di tale quadrilatero sono congruenti con il tipo di tale quadrilatero. Ad esempio, se la sequenza contenesse:
TRAPEZIO 10 10 2 5
QUADRATO 4 4 4 4
QUADRATO 3 4 3 4
RETTANGOLO 4 5 6 7
allora questa funzionalità ritornerebbe *vero*, *vero*, *falso* e *falso*, per, rispettivamente, gli indici 0, 1, 2, e 3.

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'*overflow* e l'inserimento di dati in formato errato e di stringhe troppo lunghe da *stdin*.

REGOLE

- Si può utilizzare ogni genere di manuale o di materiale didattico di altra natura
- Per superare la prova, il programma deve essere perfettamente funzionante nelle parti 1 e 2. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo (almeno 30) si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati