

## Programmazione I

### Prova di programmazione – 15 Gennaio 2018 – 2 ore

Partendo dal frammento di codice fornito, realizzate un programma per la gestione dello stoccaggio di scatole all'interno di contenitori. Ciascun contenitore è individuato da una *posizione*, rappresentata mediante un numero naturale. Il programma permette di creare una *lista di operazioni di stoccaggio*, in cui ciascuna operazione consiste nell'aggiunta di scatole all'interno di un contenitore. In particolare ciascuna operazione è caratterizzata da una coppia di informazioni:

1. posizione del contenitore di destinazione;
2. numero di scatole da stoccare.

Per semplicità assumere che **non ci sia limite** né al numero di scatole che possono essere inserite in un contenitore, né al **valore massimo delle posizioni** dei contenitori. Notare che la posizione del contenitore per una data operazione e la posizione di tale operazione nella lista in generale non coincidono (per esempio, la seconda operazione nella lista potrebbe riguardare un contenitore in posizione 40). All'avvio del programma, non esiste alcun contenitore e la lista delle operazioni è vuota. Il programma deve fornire le seguenti funzionalità.

1. **cambia\_max\_operazioni(N)** Fissa ad **N** la lunghezza massima della lista di operazioni (non limita il valore massimo delle posizioni dei contenitori!). Il valore massimo possibile per **N** non è noto a tempo di scrittura del programma. L'eventuale precedente contenuto della lista di operazioni è perso.
2. **inserisci\_operazione(p, n)** Aggiunge alla lista di operazioni una operazione di stoccaggio di **n** scatole all'interno del contenitore in posizione **p**. Se nella lista c'è già una operazione di per la posizione **p**, allora la nuova operazione viene *fusa* con quella già presente nella posizione **p**, ossia si aggiunge semplicemente **n** al numero di scatole già previsto dall'operazione preesistente.
3. **stampa\_operazioni** Stampa la lista di operazioni da effettuare. L'ordine con cui sono stampate le operazioni non è rilevante. Per ciascuna operazione stampa la posizione del contenitore di destinazione ed il numero di scatole da stoccare. Ad esempio  
30 60  
53 24  
41 10
4. **salva\_lista** Salva la lista di operazioni su di un file di testo dal nome predefinito.
5. **carica\_lista** Carica la lista di operazioni dal file. L'eventuale precedente lista è persa.
6. **inserisci\_operazione2(p, n)** Stessi passi della funzionalità 2, ma in più *fonde* la nuova operazione non solo con un'eventuale operazione già inserita per la stessa posizione, ma anche con l'operazione più vicina, tra quelle già inserite, in termini di posizione del contenitore, e solo a patto che la posizione di tale operazione non differisca da **p** per più di 20 posizioni. Ad esempio, data la lista di cui all'esempio al punto 3
  1. se **p=5**, allora non avviene alcuna fusione e l'operazione è aggiunta alla lista
  2. se **p=35**, allora l'operazione è fusa con quella già presente per la posizione 30
  3. se **p=37**, allora l'operazione è fusa con quella già presente per la posizione 41Si ottiene il punteggio massimo se si implementa questa funzionalità col numero di passi minimo per una ricerca lineare (tralasciare la ricerca binaria, che non si è vista a lezione).

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 da *stdin*.

---

#### REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. 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
  - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale