

**Programmazione I**  
**Prova di programmazione – 8 Settembre 2022 – 2 ore e mezza**

Partendo dal frammento di codice fornito, realizzare un programma per manipolare un insieme di numeri naturali. L'insieme può contenere numeri compresi tra 1 ed un valore massimo  $N$ . All'avvio del programma,  $N$  è uguale a 0, ossia l'insieme non può contenere ancora alcun numero. Il valore di  $N$  può essere modificato durante l'esecuzione del programma, come spiegato in dettaglio sotto. Assumiamo che l'insieme contenga mediamente  $O(N)$  elementi. Quindi non vi è alcun guadagno nell'implementare l'insieme con una struttura dati che permetterebbe di eseguire le varie operazioni sull'insieme in meno di  $O(N)$  passi se invece l'insieme contenesse meno di  $O(N)$  elementi. Il programma fornisce le seguenti funzionalità.

1. **[+1] inizializza\_insieme(N)** Inizializza l'insieme a contenere numeri naturali compresi tra 1 ed  $N$ . Non vi è alcun limite esplicito al valore massimo di  $N$ . L'insieme è inizializzato a vuoto, e l'eventuale precedente contenuto è perso. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
2. **unisci\_con\_insieme()** Unisce l'insieme con un secondo insieme di numeri naturali (la dimensione massima del primo insieme rimane inalterata). Il secondo insieme è letto da *stdin*, con il formato seguente. L'utente inserisce prima il numero di elementi dell'insieme, poi ciascuno dei numeri nell'insieme. Ad esempio, per inserire l'insieme di tre numeri {5, 2, 4}, all'interno di un insieme che ha dimensione massima  $N=7$ , l'utente inserisce  
**3 5 2 4**  
Lascia l'insieme inalterato se il secondo insieme è vuoto oppure contiene valori errati, o infine contiene un numero di valori non accettabile. Vediamo alcuni esempi di unioni, nel caso in cui  $N=5$  e l'insieme contenga i numeri {2, 5}. Nella prima colonna l'input dell'utente, nella seconda il nuovo contenuto che avrebbe l'insieme, con i numeri riportati in ordine crescente:  
**3 5 2 4** → {2, 4, 5}  
**3 3 2 4** → {2, 3, 4, 5}  
**0** → {2, 5}  
**5 3 2 1 5 6** → {2, 5} Inalterato perché il secondo insieme contiene un numero troppo grande
3. **stampa\_insieme** Stampa i numeri contenuti nell'insieme, separati da spazi, ed in ordine di valore crescente. Ad esempio, per l'insieme {2, 5}  
**2 5**
4. **[3] salva\_insieme** Salva l'insieme in un file **binario**.
5. **[2, +1] carica\_insieme** Carica l'insieme dal file binario. L'eventuale precedente contenuto dell'insieme è perso. Si ottiene il punteggio aggiuntivo se si evitano deallocazioni e riallocazioni di memoria non necessarie.
6. **[4] interseca\_con\_insieme** Identica alla funzionalità 2, tranne che per l'operazione effettuata sull'insieme: in questa funzionalità si effettua l'intersezione al posto dell'unione. Ripetendo gli stessi esempi del punto 2, l'insieme {2, 5} diverrebbe:  
**3 5 2 4** → {2, 5}  
**3 3 2 4** → {2}  
**0** → {}  
**5 3 1 3 6** → {2, 5} Inalterato perché il secondo insieme contiene un numero troppo grande
7. **[4] stampa\_sequenza\_massima** Stampa la sequenza più lunga di numeri consecutivi di valore crescente contenuta nell'insieme. Stampa tale sequenza separando i numeri con spazi. Ecco alcuni esempi di stampa, in funzione del contenuto dell'insieme:  
{1, 3, 4, 7, 9, 10, 11, 13} → **9 10 11**  
{2, 4, 5, 6, 7, 10, 12, 13, 14} → **4 5 6 7**

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*.

---

Per il collaudo: se fate stampare messaggi per invitare l'utente ad inserire valori, ricordate di aggiungere la stampa di caratteri accapo. Altrimenti nel puro output del programma vi saranno delle righe fuse, e di

fatto tale output non sarà quello che credete (le righe fuse non le vedete quando usate il programma da terminale, perché inserite voi l'accapo da utenti).

---

## REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1, 2 e 3. 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 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