Tecniche di Programmazione (2018/19)

Esercitazione 8

Argomento: Tipi di dato astratti

Scaricare la cartella **Esercitazione 8** che è composta da 3 sottocartelle: 1) **insiemi**, 2) **pile** e 3) **code**. In ciascuna sottocartella, completare le definizioni delle funzioni presenti nei file **esercizio_code.c**, **esercizio_insiemi.c** e **esercizio_pile.c**. Compilare e testare il tutto tramite il programma main di prova fornito che si trova all'interno di **e8_test_esercizio.c**.

Insiemi

Esercizio 8.1

Si consideri l'implementazione del tipo astratto Insieme fornita nei file **esercizio_insiemi.c** e **esercizio_insiemi.h**, dove si è scelta una rappresentazione mediante SCL ed uno schema realizzativo con side-effect senza condivisione di memoria. Le funzioni di seguito richieste devono essere implementate senza accedere direttamente alla struttura dati che rappresenta l'insieme, ma solo utilizzando le funzioni messe a disposizione dal tipo astratto.

Implementare la funzione C

Insieme* intersezione(Insieme* a, Insieme* b);

che, dati in ingresso due insiemi `a` e `b`, restituisce l'insieme corrispondente all'intersezione tra i due.

Esercizio 8.2

Implementare la funzione C

Insieme* unione(Insieme* a, Insieme* b);

che, dati in ingresso due insiemi `a` e `b`, restituisce l'insieme corrispondente all'unione dei due.

Esercizio 8.3

Implementare la funzione C

bool uguale(Insieme* a, Insieme* b);

che, dati in ingresso due insiemi `a` e `b`, restituisce true se e solo se gli insiemi a e b sono uguali.

Code

Si consideri l'implementazione (incompleta) del tipo astratto Coda, fornita nei file **esercizio_code.c** e **esercizio_code.h**, dove si è scelta una rappresentazione mediante array ed uno schema realizzativo con side-effect senza condivisione di memoria. Nell'implementazione delle seguenti funzioni scegliere la modalità di gestione dell'array che si preferisce.

Esercizio 8.4

Implementare la funzione C

Coda*codaVuota();

che restituisce un riferimento ad una coda vuota.

Esercizio 8.5

Implementare la funzione C

bool estVuota(Coda* c);

che restituisce true se e solo se la coda a cui c fa riferimento è vuota.

Esercizio 8.6

Implementare la funzione C

void inCoda(Coda* c, T e);

che inserisce l'elemento e in fondo alla coda `c`. Se l'array è pieno, riallocarlo secondo una

strategia di propria scelta.

Esercizio 8.7

Implementare la funzione C

void outCoda(Coda* c);

che estrar l'elemento in testa alla coda `c`. Scegliere autonomamente se/come ridimensionare l'array dopo l'estrazione.

Pile

Si consideri l'implementazione (incompleta) del tipo astratto Pila, fornita nei file **esercizio_pile.c** e **esercizio_pile.h**, dove si è scelta una rappresentazione mediante SCL ed uno schema realizzativo funzionale con condivisione di memoria.

Esercizio 8.8

Implementare la funzione C

Pila push(Pila p,T e);

che restituisce una nuova pila ottenuta dalla pila di input p a cui è stato aggiunto l'elemento e.

Esercizio 8.8

Implementare la funzione C

Pila pop(Pila p);

che restituisce una nuova pila corrispondente alla pila di input p privata dell'elemento affiorante.