

-- -- -- 09 Febbraio 2009 -- -- --

Laurea in Informatica

Università degli Studi di Napoli "Federico II"

Nome e Cognome

Numero di Matricola:

**Spazio riservato alla correzione**

1	2	3	4	Totale
/8	/6	/10	/10	/34

Non utilizzate altri fogli. Utilizzate soltanto lo spazio sottostante. Fogli differenti non saranno presi in considerazione per la correzione. Non scrivere a matita. Per tutti gli esercizi, descrivere la complessità asintotica delle funzioni implementate

1. Si consideri una coda **Q**, implementata con array **Q[MAX+2]**, **riempita con interi**. Si implementi la funzione ricorsiva **void moltiplica (int Q[])** che, utilizzando una libreria di funzioni di accesso alla coda (**da implementare**) prendendo in input **Q**, restituisce la coda con gli elementi nello stesso ordine, sostituendo però ogni tripla di numeri consecutivi (il "resto" è lasciato inalterato) con il loro prodotto, dando priorità agli elementi più vicini alla testa. Si ricordi che la coda è una struttura dati che permette l'accesso ai suoi dati solo dalla testa.  
Esempio: Coda iniziale 3|2|2|4|1|5|7|2 (3 testa della coda) – coda finale 12|20|7|2|

2. Si considerino due liste di numeri interi **Lista1** e **Lista2** implementate come liste doppiamente puntate e non circolari, utilizzando la seguente struttura

```
struct elemento {  
    struct elemento *prev;  
    int inf;  
    struct elemento *next;}  
  
struct elemento *Lista1,*Lista2;
```

Si implementi una funzione che preso in input un intero **el** restituisca Lista1 e Lista2 rimuova da entrambe le Liste tutte le occorrenze di **el**. Si implementi la funzione in modo **ricorsivo**.

3. Si consideri una coda di priorità per la gestione della coda di stampa di una rete implementata con una struttura dati heap **H[**MAX**]**. Si supponga di avere memorizzata la dimensione dello heap in **heapsize**. Si implementi una funzione che controlli che H sia effettivamente un heap. Si implementi poi una funzione per la soluzione del seguente problema: sia k un numero intero. Si modifichi l'i-esimo elemento dello heap aggiungendogli k, ovvero  $A[i]$  viene sostituito con  $A[i] + k$ , e di ripristinare lo heap.

4. Si considerino due grafi **G** e **H** grafi orientati pesati di **n** vertici  $0,1,\dots, n-1$  rappresentati con liste di adiacenza utilizzando la seguente struttura:

```
typedef struct graph {
    int nv;
    edge **adj; } graph;

graph *G, *H;

typedef struct edge {
    int key;
    int peso;
    struct edge *next; } edge;
```

Scrivere in linguaggio C una funzione che presi in input i grafi **G** e **H**, restituisca un nuovo grafo **T** con la stessa struttura in modo tale **T** ha l'arco  $(a,b)$  se esso è presente sia in **G** che in **H** e il suo peso è dato dalla somma del relativo peso in **G** e **H** più il grado adiacente di  $a$  più il grado incidente di  $b$  in **H**.

This document was created with Win2PDF available at <http://www.win2pdf.com>.  
The unregistered version of Win2PDF is for evaluation or non-commercial use only.  
This page will not be added after purchasing Win2PDF.