



Commento al Laboratorio n. 6

Esercizio n.1: Playlist

Si tratta di una semplice applicazione del principio di moltiplicazione. Essendo le scelte su dati semplici (stringhe per i titoli delle canzoni), la `struct amico`, oltre al numero di canzoni proposte, contiene un vettore di stringhe per i titoli, senza quindi ricorrere agli interi per rappresentare le scelte. La funzione wrapper `princMolt` alloca il vettore delle soluzioni (interi che servono da indici per il vettore di stringhe dei titoli) e la funzione del principio di moltiplicazione. La funzione `princMoltR` è standard.

Esercizio n.2: Allocazione di matrici

Le funzioni di allocazione della matrice `malloc2dR` e `malloc2dP` seguono il modello presentato in *Puntatori e strutture dati 3.3.3*. Esse sono estese per ricevere come parametro il nome del file da cui verranno letti i dati e per fornire come parametri passati per riferimento il numero di righe e di colonne. Si ipotizza di cominciare da una cella nera in alto a sinistra.

Il numero di celle bianche/nere (nw/nb) è facilmente calcolabile, dati il numero di righe nr e il numero di colonne nc :

- se almeno uno tra nr e nc è pari, allora $nw = nb = nr * nc / 2$
- se entrambi nr e nc sono dispari, visto che si inizia con una cella nera in alto a sinistra, $nw = nr * nc / 2$ e $nb = nr * nc / 2 + (nr * nc) \% 2$.

I 2 vettori `w` e `b` sono allocati dinamicamente da `separa`, che li riempie scandendo la matrice. Le celle nere sono quelle in cui la somma degli indici di riga e di colonna è pari, quelle bianche in cui è dispari.

Dopo averli stampati, il main dealloca i vettori `w` e `b`.

Esercizio n.3: Azienda di trasporti - multiordinamento

Le stringhe della `struct voce_t` sono ora dei vettori allocati dinamicamente, come pure i vettori di `voce_t log` e di puntatori `logC`, `logD`, `logP` e `logA`. Il menu viene esteso con la possibilità di leggere più volte da file (funzione `leggiTabella`), previa eliminazione della tabella corrente (funzione `liberaTabella`). Al termine la tabella corrente viene distrutta per evitare memory leak.

Nella funzione `leggiTabella` si alloca la tabella `tab` e, acquisito il numero di voci `n_voci`, vengono allocati dinamicamente i vettori `log`, `logC`, `logD`, `logP` e `logA` che sono i suoi campi. Per la lettura delle stringhe si allocano staticamente dei vettori di caratteri sovradimensionati `tmpCodice`, `tmpPart`, `tmpDest`, `tmpData`, `tmpOraP`, `tmpOraD`, i cui valori sono letti da file. Si usa `strdup` per allocare i corrispondenti campi della cella del vettore `log` e copiarvi le stringhe lette.

La funzione `liberaTabella` libera mediante `free` i vettori `logC`, `logD`, `logP` e `logA`, poi, iterando sul vettore `log`, libera i suoi campi stringa ed al termine libera il vettore `log` ed infine la tabella `tab`. Si osservi che tutto ciò che è stato allocato dinamicamente viene liberato in ordine inverso rispetto a quello di allocazione.