

Esercitazione di laboratorio n. 2

Esercizio n.1: Massim Comun Divisore

Competenze: ricorsione matematica (Ricorsione e problem-solving: 2.2)

Dati 2 interi positivi a e b , il loro massimo comun divisore ($\text{gcd}(a, b)$) si definisce ricorsivamente come:

$$\text{gcd}(a, b) = \begin{cases} 2 \cdot \text{gcd}(\frac{a}{2}, \frac{b}{2}) & \text{if } a, b \text{ are even} \\ \text{gcd}(a, \frac{b}{2}) & \text{if } a \text{ is odd, } b \text{ is even} \\ \text{gcd}(\frac{a-b}{2}, b) & \text{if } a, b \text{ are odd} \end{cases}$$

Si individui un'opportuna condizione di terminazione e si scriva una funzione ricorsiva `int gcd(int a, int b);` che realizzi la definizione di cui sopra. Si ricordi che nel calcolo del massimo comun divisore si assume che $a > b$. Per tener conto di tale condizione, se non soddisfatta, si scambiano a e b .

Esercizio n.2: Allocazione Matrici

Competenze: Strutture dati dinamiche, matrici dinamiche create da funzioni (Puntatori e strutture dati dinamiche: 3.3.3)

Un file di testo contiene una matrice di interi con il seguente formato:

- la prima riga del file specifica le dimensioni della matrice (numero di righe `nr` e numero di colonne `nc`)
- ciascuna delle `nr` righe successive contiene gli `nc` valori corrispondenti a una riga della matrice, separati da uno o più spazi.

Si scriva un programma che allochi dinamicamente la matrice ed effettui la lettura del file. La funzione di allocazione della matrice può:

- far uso del valore di ritorno per restituire il puntatore alla matrice al main `int **malloc2dR(...);`
- restituire il puntatore alla matrice tra i parametri passati per riferimento `void malloc2dP(int***, ...);`

Una volta acquisita la matrice, il programma deve invocare una funzione

`void separa(int **mat, int nr, int nc, ...)`

che:

- interpreti la matrice come se fosse una scacchiera (su ogni riga e su ogni colonna celle bianche e nere alternate)
- separi i dati delle caselle nere da quelli delle caselle bianche, copiandoli (in ordine arbitrario) in due vettori dinamici, che vanno ritornati al programma chiamante

- i due vettori di interi, di lunghezza opportuna e tale da poter contenere gli elementi delle "celle bianche" e delle "celle nere" separatamente, vanno allocati dinamicamente
- i due vettori e i loro contenuti devono essere visibili al chiamante della funzione `separa()` (andranno quindi dichiarati in modo opportuno, al posto dei puntini ...), che li stampa e li dealloca.

Esercizio n.3: Azienda di trasporti – multiordinamento

Competenze: vettori di struct, strutture dati dinamiche, vettori dinamici, (Puntatori e strutture dati dinamiche 2.5.5, 3.3.3, 3.2.3)

Si ripeta l'esercizio 4 del laboratorio 1 utilizzando vettori allocati dinamicamente. In particolare, i vettori vanno allocati in base al numero di dati effettivamente presenti nel file acquisito. Oltre ai comandi già previsti, si chiede di aggiungerne uno di acquisizione (lettura) di un nuovo file (dato il nome). Si prevede quindi non solo di acquisire una volta sola i dati da file, all'inizio dell'esecuzione, ma di poterlo fare ogni volta che l'utente lo richieda. Il comando di lettura, in generale, prevede l'acquisizione di nuovi dati da un file diverso dal precedente. Occorre quindi liberare (utilizzando la funzione `free`) i vettori precedentemente allocati, e allocarne di nuovi per i nuovi dati da leggere. Si noti che il compito di acquisizione dei dati può essere svolto da una sola funzione, chiamata sia (una prima volta) all'avvio del programma che ad ogni attivazione del comando di lettura.