

Esame di Tecniche di Programmazione

23 febbraio 2015

Istruzioni per lo svolgimento della prova:

- Inserire nel file `studente.txt` i dati personali richiesti, quindi *salvare e chiudere* il file.
- All'esercizio i è associata la directory `esercizioi`, contenente i file necessari allo svolgimento dell'esercizio. Ciascun esercizio deve essere svolto modificando *esclusivamente* i file `esercizio.h` (se necessario) ed `esercizio.c` presenti nella directory corrispondente. Tali file *non devono essere spostati* dalla directory in cui si trovano. *Nessun* altro file deve essere modificato. Non devono essere creati ulteriori file.
- Il file contenente il programma principale di ciascun esercizio è `test.c`. Tale file deve essere compilato usando il `makefile` fornito nella directory ed il corrispondente eseguibile (`test`) deve essere lanciato per eseguire i test di correttezza e conoscerne l'esito. Il file `test.c` *non deve essere modificato*.
- Verranno corretti esclusivamente gli esercizi la cui compilazione dia esito positivo. In caso contrario verrà assegnato punteggio nullo.
- È permesso consultare la documentazione della libreria standard C, raggiungibile all'indirizzo riportato nel file `link-documentazione-C.txt` (in inglese).
- Per eventuali problemi o dubbi, *chiedere al docente*.

Esercizio 1

Implementare la funzione C:

- `int* sommaColonne(int** mat, int n, int m)`

che, presa in input una matrice `mat` di interi di `n` righe ed `m` colonne rappresentata come array di array, restituisca un array di `m` elementi il cui elemento in posizione i è dato dalla somma degli elementi dell' i -esima colonna di `mat`. Si assuma che `mat` non sia vuota.

Esempio Con la seguente matrice `mat` di input:

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix}$$

L'invocazione `sommaColonne(mat,3,4)` deve restituire l'array:

$$[15 \quad 18 \quad 21 \quad 24]$$

Esercizio 2

Una *lista circolare* è una struttura collegata lineare in cui l'ultimo elemento della lista ha come successore il primo (anziché nessun elemento). Usando il tipo di dato **TipoSCL** (definito nel file **esercizio.h**) per la rappresentazione di strutture collegate lineari contenenti elementi di tipo **int**, realizzare la funzione

- `void accoda(TipoSCL* pscl, int i)`

che, presi in input un puntatore **pscl** ad una lista circolare di interi ed un intero **i**, inserisca l'elemento **i** in coda alla lista, mantenendone la circolarità.

Esempio Se la lista circolare puntata da **pscl** è $[1, 2, 3, 4]$ (dove 1 è il successore di 4), l'esecuzione di **accoda** con parametro **i** pari a 5 deve modificare la lista in $[1, 2, 3, 4, 5]$, con 1 successore di 5.

Nota Nell'output del programma di test, una lista circolare contenente, nell'ordine, gli elementi l_1, \dots, l_n è rappresentata mediante la stringa $[l_1 \rightarrow l_2 \rightarrow \dots \rightarrow l_n \rightarrow | l_1]$, dove l'elemento dopo il carattere **|** indica il successore dell'ultimo elemento della lista (ovvero il primo elemento della lista).

Esercizio 3

Implementare la funzione **ricorsiva C**:

- `int sommatoria(int n, int m)`

che, presi in input due interi strettamente positivi **n** ed **m**, restituisca il valore: $\sum_{i=1}^m n^i = n + n^2 + \dots + n^m$. **Non è consentito l'uso di istruzioni di ciclo, né nel corpo della funzione né in eventuali funzioni ausiliarie.**

Esempio L'invocazione **sommatoria(2,4)** deve restituire il valore $2 + 2^2 + 2^3 + 2^4 = 2 + 4 + 8 + 16 = 30$.