

Laboratorio: memoria dinamica

Stefano Ferrari

Università degli Studi di Milano
stefano.ferrari@unimi.it

Programmazione

anno accademico 2017–2018

Vettore dinamico

- ▶ Creare un vettore dinamico di dimensione N data dall'utente
- ▶ Mettere nei suoi elementi tutti i numeri da 1 a N
- ▶ Stamparli a video
- ▶ Note:
 - ▶ si può usare sia `malloc`, sia `calloc`
 - ▶ controllare il valore restituito dalla funzione di allocazione
 - ▶ è `NULL`?
 - ▶ se N è troppo elevato, la stampa può essere fatta solo per alcuni valori
 - ▶ i primi e gli ultimi elementi
 - ▶ ogni tot elementi
 - ▶ solo tot elementi equidistribuiti
- ▶ raffinamento (dopo aver fatto quanto sopra):
 - ▶ N non può essere più grande del massimo intero rappresentabile (altrimenti non possiamo scriverne il valore nel corrispondente elemento del vettore): è possibile includere `limits.h` e controllare che N non sia maggiore di `INT_MAX`

Vettore di puntatori

- ▶ Predisporre un array di N puntatori a `int`
 - ▶ N può essere definito da una macro
- ▶ Allocare nella memoria dinamica N interi
 - ▶ referenziati da un elemento del vettore di puntatori
- ▶ Mettere negli interi creati tutti i numeri da 1 a N
- ▶ Stamparli a video

Matrice “quasi” dinamica

- ▶ Predisporre un array di N puntatori a `int`
 - ▶ N può essere definito da una macro
- ▶ Allocare per ogni elemento del vettore di puntatori un vettore di M interi
 - ▶ M può essere definito da una macro
- ▶ Mettere nel vettore referenziato dal puntatore nell'elemento i i primi M multipli di $i + 1$
- ▶ Stampare a video la matrice così costruita

Matrice dinamica

- ▶ Ripetere l'esercizio precedente (matrice "quasi" dinamica) partendo solo da un puntatore a puntatore di `int`
- ▶ Esso va usato per referenziare un vettore di N puntatori a `int` e per ciascuno di essi va allocato un vettore di M interi
 - ▶ N e M possono essere definiti da una macro
- ▶ Mettere nel vettore referenziato dal puntatore nell'elemento i i primi M multipli di $i + 1$
- ▶ Stampare a video la matrice così costruita
- ▶ Deallocare la memoria partendo dai vettori di `int` e solo dopo rilasciare la memoria referenziata dal puntore di puntatore.