

LABORATORIO DI PROGRAMMAZIONE 1
CORSO DI LAUREA IN MATEMATICA
UNIVERSITÀ DEGLI STUDI DI MILANO
2018–2019
29.V.2019

VINCENZO MARRA

INDICE

Esercizio 1	2
<i>Ordinamento BubbleSort con allocazione dinamica</i>	2
Tempo: 25 min.	2
Esercizio 2	2
<i>Ordinamento BubbleSort con allocazione dinamica, menu, e funzioni</i>	2
Tempo: 35 min.	2
Esercizio 3	2
<i>Ordinamento BubbleSort con riallocazione dinamica</i>	2
Tempo: 30 min.	3
Esercizio 4	3
<i>Ordinamento di double su file</i>	3
Tempo: 30 min.	3

ESERCIZIO 1

Ordinamento BubbleSort con allocazione dinamica.

Tempo: 25 min.

Si scriva un programma che legga dalla console una successione di valori `double`, e li memorizzi in una zona della memoria di dimensione appropriata. Usate le funzioni di `stdlib.h` per l'allocazione dinamica della memoria. Per convertire i valori passati dalla riga di comando in `double` potete usare la funzione `double atof(const char *str)` della libreria standard. Il programma ordina poi gli elementi della lista applicando l'algoritmo `BubbleSort` (già visto in laboratorio), visualizza l'elenco ordinato e termina.

ESERCIZIO 2

Ordinamento BubbleSort con allocazione dinamica, menu, e funzioni.

Tempo: 35 min.

Si scriva un programma che visualizzi un menu come segue:

1. Inserisci elenco di `double`.
2. Ordina elenco.
3. Visualizza elenco.
4. Termina.

Se l'utente sceglie 4 il programma termina. Se l'utente sceglie 2 o 3 prima di aver inserito un elenco di `double`, il programma informa l'utente della necessità di inserire dei dati e torna al menu. Se l'utente sceglie 1 il programma permette all'utente di inserire un elenco di valori di tipo `double`, e li memorizza usando le funzioni di `stdlib.h` per l'allocazione dinamica della memoria. Potete decidere di chiedere preliminarmente all'utente quanti valori intende inserire, oppure di lasciare che l'utente inserisca i valori liberamente fino a quando non digita direttamente invio. (Notate però che la prima implementazione è più semplice, perché basta una sola chiamata a `malloc`.) Stabilite un comportamento sensato del programma nel caso in cui l'utente scelga 1 e inserisca, o tenti di inserire, zero valori. L'opzione 3 permette di visualizzare l'elenco corrente. L'opzione 2 ordina l'elenco corrente applicando l'algoritmo `BubbleSort`. Se l'utente sceglie 1 dopo aver già inserito un elenco, il programma scarta il vecchio elenco (deallocando la memoria con `free` di `stdlib.h`) e permette all'utente di inserirne uno nuovo, allocando la memoria necessaria di conseguenza. Strutturate il programma in funzioni appropriate — per esempio, funzioni per l'input dei dati e per l'implementazione di `BubbleSort`.

ESERCIZIO 3

Ordinamento BubbleSort con riallocazione dinamica.

Tempo: 30 min.

Si modifichi il programma scritto per risolvere l'Esercizio 2 di modo che il menu diventi:

1. Inserisci elenco di `double`.
2. Ordina elenco.
3. Visualizza elenco.
4. Aggiungi elementi.
5. Termina.

L'opzione 4 permette all'utente di aggiungere valori `double` in coda all'elenco di valori corrente. Come nell'esercizio precedente, potete decidere di chiedere preliminarmente all'utente quanti valori intende aggiungere all'elenco corrente, o di lasciare che l'utente inserisca i valori liberamente fino a quando non digita direttamente `invio`. Usate `realloc` di `stdlib.h`. Se l'utente sceglie 4 prima di aver inserito un elenco di `double` tramite 1, il programma informa l'utente della necessità di inserire dei dati e torna al menu. Per il resto il funzionamento del programma è come nell'Esercizio 2.

ESERCIZIO 4

Ordinamento di `double` su file.

Tempo: 30 min.

Scrivete un programma che riceva due nomi di file dalla riga di comando. Il programma apre il primo file in lettura, e assume che esso sia formattato come un elenco di valori `double`, scritti uno per riga. Il programma legge i valori `double` usando la funzione `fscanf` di `stdio.h`, e li memorizza usando l'allocazione dinamica della memoria. (Potete prevedere una funzione `add`, di prototipo appropriato, che esegua il compito di aggiungere alla memoria ciascun nuovo valore `double` passato in argomento.) Il programma esegue poi un ordinamento dell'elenco di numeri, usando per esempio `BubbleSort`. Il programma infine scrive sul secondo file ricevuto dalla riga di comando la lista ordinata, usando `fprintf` di `stdio.h`.

(V. Marra) DIPARTIMENTO DI MATEMATICA *Federigo Enriques*, UNIVERSITÀ DEGLI STUDI DI MILANO, VIA CESARE SALDINI, 50, I-20133 MILANO

Email address: `vincenzo.marra@unimi.it`