



NON CONSEGNARE
LA BRUTTA COIA

CONSEGNARE ANCHE
QUESTO FOGLIO

Cognome	Nome				Matricola			Voto: .../30
---------	------	--	--	--	-----------	--	--	--------------

Quesito:	1	2	3	4			Tot.
Max:	8	7	10	5			30
Punti:							

Istruzioni:

- Per superare la prova è necessario conseguire almeno 18/30;
- non è consentito consultare libri, appunti, la calcolatrice o qualsiasi dispositivo elettronico, né comunicare;
- la chiarezza, sia grafica sia espositiva, costituisce un criterio significativo nella valutazione della prova
- tempo a disposizione: 1h 40m.

Stile del codice C:

- i commenti non sono necessari, ma potrebbero essere utili in caso di errore;
- è interesse dello studente assegnare a variabili, costanti e parametri nomi che ne rendano chiaro l'utilizzo nel programma.

Quesito 1. (8 punti). Scrivere un sottoprogramma che, ricevendo in ingresso un float con valore compreso tra 0 e 1 (esclusi gli estremi), generi le prime 23 cifre della sua rappresentazione binaria. Dato il numero frazionario 1/3 (un terzo), calcolarne la rappresentazione floating point a singola precisione secondo il formato IEEE 754.

Quesito 2 (7 punti). Si scriva un sottoprogramma *ricorsivo*, avente prototipo **void RicBubbleSort (float Arr[], int N)**, che effettui l'ordinamento mediante l'algoritmo di BubbleSort* (ordinamento a bolle) di un array di float ricevuto in ingresso.

Quesito 3. (10 punti). Si scriva un sottoprogramma che, ricevendo in ingresso una lista dinamica di (almeno due) valori reali, la ordini in ordine crescente mediante l'algoritmo BubbleSort*.

Suggerimento: è conveniente, ad ogni iterazione, portare il valore massimo in fondo alla lista anziché il valore minimo in testa alla lista.

Nota: la versione implementata dell'algoritmo BubbleSort non deve essere necessariamente la più efficiente.

Quesito 4. (5 punti). Si consideri un file binario contenente dati di tipo Dato. Il tipo Dato è una struct composta da due campi: un campo intero chiamato Valore e un campo carattere chiamato Chiave. Il file contiene uno o più di questi dati.

Si scriva un programma C che:

1. apre un file binario del tipo sopra descritto, situato nella directory corrente, il cui nome è "Archivio";
2. legge le struct contenute nel file alla ricerca di quella (se presente) il cui campo Chiave vale '#';
3. se tale struct esiste, il programma modifica la prima delle struct contenute nel file, copiando nel suo campo Valore il campo Valore della struct il cui campo Chiave vale '#' (senza modificare le struct successive contenute nel file);
4. chiude il file.

Se il file non contiene alcuna struct il cui campo Chiave vale '#', il programma lascia inalterato il file. Se il file contiene più struct il cui campo Chiave vale '#', solo la prima di esse viene presa in considerazione.

Ogniqualvolta esegue operazioni di apertura o chiusura file, il programma deve segnalare (con appositi messaggi a schermo) eventuali errori riscontrati nel loro svolgimento.

Nota: aprendo un file in modalità lettura+scrittura, una operazione di scrittura eseguita in testa al file sovrascrive solo il dato situato in testa al file, senza alterare la parte restante del file.

- In caso di necessità è possibile prendere visione presso il docente dell'algoritmo BubbleSort con una penalità di 3 punti: i 3 punti verranno sottratti al punteggio complessivo degli esercizi 2 e 3