

Esercizio 1

Scrivere un programma che utilizzi la procedura *qsort* e ordini un vettore di interi non negativi (in input), in modo da ottenere il seguente effetto. L'array riordinato deve contenere prima tutti i numeri pari e, a seguire, i numeri dispari. I numeri pari devono essere ordinati in modo crescente fra di loro. I numeri dispari devono essere ordinati in modo decrescente fra di loro. La sequenza in input è, come al solito, il numero N (non limitato) di interi seguiti da N valori interi non negativi. Stampare la sequenza riordinata in output su una sola riga. *Nota*: il numero zero è, per convenzione, ritenuto sempre pari.

Esercizio 2

Scrivere un programma che legga da tastiera un array A di stringhe e che utilizzi la funzione di libreria *qsort* per ordinare in ordine alfabetico crescente le stringhe in input. Stampare in output la sequenza di stringhe ordinata, una per riga. L'input è formattato nel seguente modo: sulla prima riga si trova il numero N di stringhe. Seguono N righe contenenti ognuna una stringa dell'insieme da ordinare. Le stringhe contengono soltanto caratteri alfanumerici ($a - z$ minuscoli e maiuscoli o numeri, nessuno spazio o punteggiatura) e sono lunghe al più 1000 caratteri ciascuna.

Esercizio 3

Scrivere un programma che implementi la ricerca binaria (dicotomica) su un insieme di stringhe in input. L'input fornito è formato da una prima riga contenente N il numero di stringhe tra cui effettuare la ricerca. Le successive N righe contengono stringhe ordinate in modo alfabetico crescente. Segue una sequenza di dimensione non conosciuta di coppie di righe. La prima riga di ogni coppia è il valore "1" o "0". Se il valore è 0 il programma termina (non ci sono più richieste). Se il valore è 1, sulla riga successiva trovate la stringa da cercare. Per tanto l'input contiene le $N + 1$ righe dell'insieme e ogni successiva richiesta di ricerca viene preceduta da una riga con il numero "1". La procedura termina quando si trova uno "0". Le stringhe contengono solo caratteri alfanumerici e non sono più lunghe di 1000 caratteri. L'output è composto da una riga per ognuna delle richieste. Ogni riga contiene il risultato della ricerca binaria: se l'elemento appare nell'insieme, stamparne la posizione (valore da 0 a $N - 1$). Altrimenti, stampare "-1".

Esercizio 4

Scrivere un programma che legga una sequenza binaria in input da tastiera e la inserisca in un array. Dopodichè utilizzare una soluzione di **divide-et-impera** ricorsiva che segnali se la sequenza contiene più 1 che 0, più 0 che 1 o se sono uguali. In particolare, nel primo caso scrivere la stringa '1\n', nel secondo, '0\n', nel terzo, '= \n'. L'input è formattato nel seguente modo: sulla prima riga si trova la dimensione N della stringa. Nessuna assunzione viene fatta sulla dimensione di N . Sulla linea seguente, la stringa binaria (senza separatori) di N bit 0 o 1.