

## Esercizio 1

Implementare l'**Insertion Sort** su interi. Scrivere un programma che utilizzi la funzione per ordinare un array  $A$  di  $N$  interi. 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 ordinata in output su una sola riga.

## Esercizio 2

Implementare l'**Insertion Sort** per ordinare in ordine alfabetico un array di stringhe. Scrivere un programma che legga da tastiera un array  $A$  di  $N$  stringhe e che utilizzi la funzione appena scritta per ordinarle. Stampare in output la sequenza di stringhe ordinata su una sola 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. Si può assumere che la lunghezza massima di ciascuna stringa sia di 100 caratteri. Utilizzare la funzione `strcmp` in `string.h` per il confronto tra stringhe. Utilizzare `man strcmp` per maggiori informazioni.

## Esercizio 3

Scrivere un programma che implementi la **ricerca binaria** 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 100 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

Implementare il **MergeSort** su interi. Scrivere un programma che utilizzi la funzione per ordinare un array  $A$  di  $N$  interi. 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 ordinata in output su una sola riga.