

Prova scritta del 8/1/2007

1) (a) Realizzare una funzione booleana di nome `SOTTOSTRINGA` che, presi come suoi parametri una stringa `S` e due caratteri `I` ed `F`, determina la prima sottostringa `SS` di `S` delimitata, a sinistra e a destra, rispettivamente dai caratteri `I` ed `F`. In particolare, se `I` è il carattere speciale `'\0'`, la stringa `SS` inizia dal primo carattere di `S`. Ad esempio, data la stringa `"xxx.yyy@studenti.unipr.it"` se `I = '\0'` ed `F = '@'`, `SS` sarà `"xxx.yyy"`, mentre se `I = '@'` ed `F = '.'`, `SS` sarà `"studenti"`. N.B. I caratteri delimitatori non fanno parte della sottostringa `SS`. Nel caso in cui non si trovi in `S` uno dei due delimitatori la funzione restituisce `false`, mentre restituisce `true` in tutti gli altri casi. N.B. Dichiarare e trattare `S` ed `SS` come stringhe del C, e cioè come array di caratteri terminati da `'\0'`. Sugg.: dichiarare `SS` (oltre che `S`, `I`, ed `F`) come parametro della funzione.

(b) Descrivere la funzione `SOTTOSTRINGA` anche tramite un diagramma di flusso.

2) Scrivere un programma principale che permetta di eseguire semplici operazioni su indirizzi di posta elettronica della forma `nome.cognome@dom1...domn` memorizzati su file (si assuma che gli indirizzi siano tutti sintatticamente corretti e di lunghezza massima 64, e siano separati l'uno dall'altro da esattamente un carattere `'\n'`). Precisamente, il programma permette di eseguire ripetutamente le seguenti operazioni:

1. caricare in un array `A` tutti gli indirizzi di posta elettronica contenuti in un file di nome specificato dall'utente (si assuma che il file possa contenere al massimo 1000 indirizzi). SUGG.: si dichiari `A` come un array bidimensionale `1000 x 64`.
2. stampare i nominativi (cognome e nome) relativi a tutti gli indirizzi memorizzati in `A` in cui appaia la stringa `"studenti"` come primo dominio (= sottostringa a destra di `@`). Si richiede di utilizzare, obbligatoriamente, la funzione `SOTTOSTRINGA` realizzata nell'esercizio (1).
3. salvare l'array `A` sul file specificato nell'operazione 1.
4. smettere.

Il programma presenta all'utente (su standard output) il menù delle possibili operazioni, esegue l'operazione scelta e quindi ripete dall'inizio finché non viene scelta l'operazione 4

3) Scrivere una funzione di nome `DISGIUNGI` che, presi come suoi parametri un'array di interi `I`, il numero `n` di elementi in `I` e due `struct I1` ed `I2`, costituite ciascuna da 3 campi, `C1`, `C2` e `C3`, rispettivamente di tipo intero, carattere e array di interi (di dimensione mass. 100), suddivide gli elementi di `I` nelle due strutture `I1` ed `I2` nel modo seguente: se l'elemento di `I` è pari viene copiato nella prima posizione libera dell'array `C3` di `I1`, altrimenti, in `C3` di `I2`. Nel campo `C1` di `I1` e di `I2` viene inserito il numero di elementi contenuti nel corrispondente campo `C3`, mentre nel campo `C2` di `I1` e di `I2` viene inserito un carattere che specifica se il corrispondente campo `C3` contiene numeri pari (`'p'`) o dispari (`'d'`).

Scrivere anche un programma principale che richiami la funzione `DISGIUNGI` passandole come parametro l'array di interi `I = {4, 6, 13, 2, 5}` e quindi stampi le due strutture dati `struct` risultanti. Mostrare la stampa prodotta.