

## Prova scritta del 12/9/2006

1) (a) Realizzare una funzione di nome `fondi` che, presi due array `A` e `B` di numeri reali, rispettivamente di dimensioni `n` ed `m`, ordinati in modo non-decrescente, fonde gli elementi dei due array in un unico array `C` mantenendo l'ordinamento non-decrescente. N.B. Si richiede di dichiarare `A`, `B`, `C`, `n`, `m` come parametri della funzione.

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

2) Scrivere un programma principale che permetta di fondere due file ordinati di numeri reali in un unico nuovo file ordinato. Precisamente, il programma procede nel modo seguente: richiede all'utente il nome dei due file da fondere (lunghezza max nome file: 32 caratteri); legge e memorizza il contenuto dei due file in due array rispettivamente di nome `sequenza1` e `sequenza2` (ciascuno di dimensione massima 1000); applica la funzione `fondi` ai due array letti, specificando di memorizzare il risultato della fusione nell'array `sequenzaTot`; memorizza l'array `sequenzaTot` in un nuovo file il cui nome è ottenuto concatenando i nomi dei due file di input. Si richiede che il programma controlli anche l'esistenza dei due file di input, terminando immediatamente l'esecuzione in caso di errore.

3) Sia `S` il tipo di una struttura dati (`struct`) costituita da 3 campi di tipo intero `X1`, `X2`, `X3` e si consideri la seguente relazione d'ordine tra due elementi qualsiasi `a` e `b` di tipo `S`: `a` precede `b` se e solo se la somma dei campi di `a` è minore della somma dei campi di `b`. Modificare la funzione `fondi` dell'esercizio (1) in modo che possa effettuare la fusione di array di elementi di tipo `S` (invece che di numeri reali). Si richiede di definire (obbligatoriamente) una funzione di nome `precede` che realizza la relazione di precedenza tra due elementi di tipo `S` sopra indicata, e di utilizzarla in modo opportuno all'interno della funzione `fondi`.

4) Scrivere un programma principale che esegue ripetutamente le seguenti operazioni: legge un carattere che rappresenta un giorno della settimana secondo la codifica sotto riportata; legge un intero che rappresenta un'ora del giorno tra le 8 e le 20; legge un altro intero e lo memorizza in una matrice di interi `M`,  $13 \times 7$ , nell'elemento di indici  $i,j$  dove  $i$  corrisponde all' $(i+1)$ -esima ora del giorno a partire dalle ore 8 e  $j$  corrisponde all' $(j+1)$ -esimo giorno della settimana a partire da lunedì. Ad esempio, dato l'input 'g', 13, 35, l'intero 35 verrà memorizzato in `M[3][5]`. Quando il carattere letto è il carattere 'x' o 'X' il programma stampa la matrice letta, un riga di stampa per ogni riga della matrice, e quindi termina.

Codifica giorni della settimana:

'l' o 'L'	lunedì
'm' o 'M'	martedì
'e' o 'E'	mercoledì
'g' o 'G'	giovedì
'v' o 'V'	venerdì
's' o 'S'	sabato
'd' o 'D'	domenica