

Prova scritta del 11/7/2006

1) (a) Realizzare una funzione di nome `PariDispari` che, presi un array di interi non negativi `A` ed il numero `nA` di elementi in `A`, suddivide gli elementi di `A` in due array `B` e `C`, copiando in `B` gli elementi con valore pari ed in `C` quelli con valore dispari, restituendo in `nB` ed in `nC` il numero di elementi copiati in `B` e `C`, rispettivamente. Nel caso in cui uno degli elementi di `A` sia un numero negativo la funzione termina immediatamente restituendo come suo valore `-1`, altrimenti la funzione restituisce `0`. N.B. Si richiede di dichiarare `A`, `B`, `C`, `nA`, `nB`, `nC` come parametri della funzione, usando il passaggio parametri per riferimento ove opportuno.

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

2) Scrivere un programma principale che permetta di suddividere il contenuto di un file di numeri interi in due file distinti, costituiti uno dai numeri pari e l'altro dai numeri dispari presenti nel file originario. Precisamente il programma procede nel modo seguente: richiede all'utente il nome del file da analizzare; legge e memorizza il contenuto del file in un array di interi `Numeri` (di dimensione massima `1000`); applica la funzione `PariDispari` all'array `Numeri`, richiedendo di memorizzare i numeri pari e dispari negli array `numeriP` e `numeriD` rispettivamente; se la funzione restituisce come suo risultato `-1`, allora il programma termina immediatamente con opportuno messaggio di errore; altrimenti, il programma memorizza i due array `numeriP` e `numeriD` su due nuovi file i cui nomi sono ottenuti dal nome del file di input, aggiungendo il suffisso `"_pari"` e `"_dispari"`, rispettivamente. Si richiede anche che il programma provveda a memorizzare in testa ai due file di output (e cioè come primo dato scritto sul file) il numero di elementi che verranno memorizzati sul file stesso.

3) (a) Sia `S` il tipo di una struttura dati (`struct`) costituita da 6 campi: `C` di tipo carattere, `X1`, `X2`, ..., `X5` di tipo intero. Scrivere una funzione di nome `Vocali` che, preso come suo (unico) parametro una struttura `V` di tipo `S`, analizza il campo `C` di `V` e se il valore di `C` è una delle 5 vocali, `'a'`, `'e'`, `'i'`, `'o'`, `'u'`, incrementa di un'unità il campo `X1`, `X2`, ..., `X5` rispettivamente (se `C` non è una vocale la funzione termina senza modificare i valori dei campi `X1`, `X2`, ..., `X5`).

(b) Scrivere anche un programma principale che esegue ripetutamente le seguenti operazioni: legge da standard input un carattere, lo memorizza nel campo `C` di una struttura `D` di tipo `S`, ed utilizzando (obbligatoriamente) la funzione `Vocali`, aggiorna i campi `X1`, `X2`, ..., `X5` di `D`. Il programma termina quando il carattere letto è il carattere `'.'`, e quindi provvede a stampare le percentuali di ciascuna vocale rispetto al numero totale dei caratteri letti così come ricavabili dai dati contenuti nella struttura `D`.