

Laboratorio di Informatica L-A (A.A. 2006/2007) - CdL Ingegneria Informatica
Prof. Zannoni – Esercitazione con consegna del 09/11/2006 - durata 2h

ESERCIZIO

Il fornitore di una cartoleria invia ogni mese un elenco cartaceo contenente i prodotti di cancelleria che può consegnare. Tali prodotti sono tutti identificati tramite un intero, unico per ogni prodotto. L'elenco cartaceo presenta i codici in maniera non ordinata.

Si vuole realizzare un programma di ausilio per la cartoleria: tale programma deve richiedere al responsabile di inserire i codici dei prodotti che è possibile ordinare (ovvero la lista del fornitore), e poi deve richiedere l'inserimento dei codici dei prodotti presenti in cartoleria. Il programma deve stampare a video l'elenco dei prodotti che la cartoleria è effettivamente interessata a ordinare, ovvero tutti i prodotti presenti nella lista del fornitore ma non in cartoleria.

A tal scopo si realizzi:

- un primo modulo *gestioneProdotti.c* (con il corrispettivo header file), che contenga le seguenti funzione e procedura:
 - **leggi (. . .)** che riceva in ingresso un array di interi, la sua dimensione massima, ed un intero passato per riferimento, dove memorizzare la dimensione logica del vettore, ovvero il numero di prodotti effettivamente letto (all'inizio il vettore è vuoto). La funzione deve chiedere all'utente di inserire un codice di un prodotto, e controllare che tale codice sia maggiore di 0. Qualora il codice sia minore di 0, la funzione deve provvedere a chiedere il re-inserimento; se invece il codice inserito è pari a 0, la funzione deve terminare, restituendo tramite il parametro passato per riferimento, il numero di prodotti letti, e tramite il parametro di ritorno un codice di successo. La funzione deve anche controllare che l'array passato per parametro abbia dimensione fisica maggiore di 0; in caso contrario, la funzione non deve chiedere nulla all'utente, ma si deve limitare a restituire un opportuno codice di errore.
 - **stampa (. . .)** che ricevuti in ingresso un array di interi rappresentanti i codici e la dimensione (logica) dell'array, stampi a video tali codici. La procedura deve essere realizzata in maniera ricorsiva (suggerimento: un array di dimensione n può essere visto come un elemento - quello di posizione zero - più un array di dimensione n-1; la condizione di terminazione è l'aver ricevuto in ingresso un array di dimensione logica 0).
- un secondo modulo *gestioneErrori.c* (con il corrispettivo header file), in cui siano definiti tutti i necessari codici di errore, ed una funzione di gestione degli errori. Tale funzione riceve come parametro un codice, che può rappresentare sia il successo che un possibile errore. Nel caso in cui il codice rappresenti un errore, la funzione deve provvedere a stampare una spiegazione di tale errore.
- un programma (in un file *main.c*) che, utilizzando la funzione **leggi (. . .)**, provveda a leggere da standard input l'elenco dei prodotti del fornitore (si supponga per semplicità che al massimo vi siano MAX_DIM prodotti, con MAX_DIM costante simbolica). Dopodiché il programma provveda a leggere anche la lista di prodotti presenti in cartoleria (ancora si supponga che non possano essere più di MAX_DIM). Qualora l'invocazione di tale funzione non abbia successo, il programma deve stampare a video il motivo del fallimento, e poi terminare immediatamente. Il programma provveda a copiare in un array di dimensioni opportune i soli prodotti che sono da ordinare, ovvero quelli presenti nella lista del fornitore ma non in cartoleria. Utilizzando poi la funzione **stampa (. . .)**, si stampi a video tale lista.

N.B. Si cerchi di inquadrare bene il problema: quale operazione bisogna eseguire sui due vettori letti per produrre il vettore da stampare?

Soluzione

main.c

```
/******  
*      Author   : Marco Montali  
*      Created  : 9/11/2006 - 8:57  
*      Modified : 9/11/2006 - 8:57  
*  
*      Description :  
*      Esercitazione con consegna del 9 novembre  
*****/  
  
#include <stdio.h>  
#include <string.h>  
#include <stdlib.h>  
#include "gestioneErrori.h"  
#include "gestioneProdotti.h"  
  
#define MAX_DIM 128  
#define TRUE 1  
#define FALSE 0  
#define BOOLEAN int  
  
int main(void)  
{  
    int dimFornitore, dimNegozio, dimOrdine=0;  
    int tempResult, i, j;  
    int prodottiFornitore[MAX_DIM];  
    int prodottiNegozio[MAX_DIM];  
    int ordine[MAX_DIM];  
    BOOLEAN daOrdinare;  
  
    tempResult = leggi(prodottiFornitore, MAX_DIM, &dimFornitore);  
    handleError(tempResult);  
  
    tempResult = leggi(prodottiNegozio, MAX_DIM, &dimNegozio);  
    handleError(tempResult);  
  
    for (i = 0; i < dimFornitore; i++)  
    {  
        daOrdinare = TRUE;  
        for (j = 0; j < dimNegozio; j++)  
        {  
            if (prodottiFornitore[i] == prodottiNegozio[j])  
            {  
                daOrdinare = FALSE;  
                break;  
            }  
        }  
        if(daOrdinare)  
        {  
            ordine[dimOrdine] = prodottiFornitore[i];  
            dimOrdine++;  
        }  
    }  
  
    stampa(ordine, dimOrdine);  
  
    system("PAUSE");  
    return (0);  
}
```

gestioneProdotti.h

```
#include <stdio.h>
#include "gestioneErrori.h"

RESULT_TYPE leggi(int codProdotti[], int maxDim, int * logicDim);
void stampa(int codProdotti[], int dim);
```

gestioneProdotti.c

```
#include "gestioneProdotti.h"

RESULT_TYPE leggi(int codProdotti[], int maxDim, int * logicDim)
{
    int temp;
    if (maxDim > 0)
    {
        *logicDim = 0;
        do
        {
            printf("Inserire il codice di un prodotto (0 per terminare): ");
            scanf("%d", &temp);
            if (temp != 0)
            {
                if (temp < 0)
                    printf("Codice inesistente!\n");
                else
                {
                    codProdotti[*logicDim] = temp;
                    *logicDim = *logicDim + 1;
                }
            }
        }
        while ((temp != 0) && (*logicDim < maxDim));
        return RESULT_OK;
    }
    else
    {
        return RESULT_WRONG_DIMENSION;
    }
}

void stampa(int codProdotti[], int dim)
{
    if(dim == 0)
        return;
    else
    {
        printf("%d\n", codProdotti[0]);
        stampa(codProdotti+1, dim-1);
    }
}
```

gestioneErrori.h

```
#include <stdio.h>

#define RESULT_TYPE int
#define RESULT_OK 1
#define RESULT_WRONG_DIMENSION -1

void handleError(RESULT_TYPE errorCode);
```

gestioneErrori.c

```
#include "gestioneErrori.h"

void handleError(RESULT_TYPE errorCode)
{
    switch(errorCode)
    {
        case RESULT_OK:
            printf("Lettura completata con successo\n");
            break;
        case RESULT_WRONG_DIMENSION:
            printf("Problemi con la dimensione del vettore\n");
            break;
        default:
            printf("Codice di errore non trattato\n");
    }
}
```