

ELEMENTI DI INFORMATICA L

PROVA D'ESAME

Files, Strutture

ESERCIZIO n.1

È dato un file di testo che contiene l'elenco dei vestiti presenti in un negozio. Il file contiene al massimo 20 righe e ogni riga contiene:

- Il tipo (un codice numerico che indica il tipo di vestito)
- La taglia (numero intero)
- La quantità di vestiti di tale tipo e taglia presenti in negozio (numero intero)
- Il prezzo unitario (floating point)

Si chiede di scrivere un programma C, denominato **vestiti**, con le seguenti caratteristiche:

Il programma dovrà essere suddiviso in 3 file.

1. **funzioni.h**: dovrà contenere le definizioni dei tipi e le dichiarazioni delle funzioni sotto descritte
2. **funzioni.c**: dovrà contenere le definizioni delle funzioni dichiarate in **funzioni.h**
3. **main.c**: dovrà contenere solo la funzione **main()**

Il programma dovrà comprendere:

Tipi di strutture

1. **vestito** che contiene i dati di un vestito (tipo, taglia e quantità)
2. **vestiti** che contiene un array di strutture di tipo **vestito** e il numero di vestiti effettivamente letti.

Funzioni e procedure

1. **funzione** denominata **leggi()**: dato il **nome di un file** riempie una struttura di tipo **vestiti** contenente le righe lette dal file. La funzione restituisce 1 se l'operazione è andata a buon fine, altrimenti restituisce 0.
2. **procedura** denominata **estrai()**: prende come parametri una taglia, e due strutture di tipo **vestiti**. La prima struttura contiene i vestiti letti con la funzione **leggi()**, nella seconda struttura bisogna ricopiare tutti i vestiti della taglia passata come primo parametro
3. **funzione** denominata **quant()** che prende come parametro una struttura di tipo **vestiti** e restituisce come risultato il numero totale di vestiti contenuti nella struttura passata come parametro
4. **funzione** denominata **prezzo()** che prende come parametro una struttura di tipo **vestiti** e restituisce come risultato il prezzo totale dei vestiti contenuti nella struttura passata come parametro
5. **funzione** denominata **scrivi()** che scrive in un file di testo, il cui **nome** viene passato come parametro, una riga per ogni taglia **pari** compresa fra il 42 e il 54. La funzione restituisce 1 se l'operazione è andata a buon fine, altrimenti restituisce 0.

Per ogni riga avremo:

- La taglia

- La quantità di vestiti della taglia in questione
- Il prezzo totale dei vestiti della taglia in questione

Attenzione: le taglie dei vestiti sono solo numeri pari (46, 48 ecc.)

N.B.: In pratica la funzione `scrivi()` per ogni taglia invocherà prima la funzione `estrai()` che estrarrà tutti i vestiti della taglia richiesta inserendoli nella struttura passata come ultimo parametro. Quindi invocherà `quant()` e `prezzo()`, passando come parametro la struttura riempita da `estrai()`.

Il programma, utilizzando le suddette funzioni, dovrà:

1. Leggere il file **VESTITI.TXT** - usando la funzione `leggi()`) segnalando a video eventuali problemi
2. Scrivere nel file di testo **TOTALI.TXT** le quantità e i prezzi totali per ogni taglia, usando la funzione `scrivi()`, segnalando a video eventuali problemi.

Soluzione

```

/* ----- */
/* Programma: vestiti      Modulo: funzioni.h      */
/* ----- */

/* ----- */
/* Definizione di costanti per rendere parametrico il programma      */
/* ----- */

#define MAXRIGHE 20

#define FALSE 0
#define TRUE 1

/* ----- */
/* 1. Definizione struttura riga      */
/* ----- */

typedef struct
{
    int tipo, taglia, quant;
    float prezzo;
} vestito;

/* ----- */
/* 2. Definizione struttura log      */
/* ----- */

typedef struct
{
    vestito v[MAXRIGHE];
    int num;
} vestiti;

/* ----- */

```

```

/* 3. Dichiarazione funzioni */
/*-----*/

int leggi(char nomefile[], vestiti* vs);
void estrai(int taglia, vestiti vs, vestiti *es);
int quant(vestiti vs);
float prezzo(vestiti vs);
int scrivi(char nomefile[], vestiti vs);

void errore(int n, char msg[]);

/*-----*/
/* Programma: vestiti      Modulo: funzioni.c */
/*-----*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "funzioni.h"

/*-----*/
/* 1. Definizione funzione leggi() */
/* Legge da nomefile i dati delle righe nella struttura vestiti */
/*-----*/

int leggi(char nomefile[], vestiti* vs)
{
    vestito v;
    FILE* f;

    f = fopen(nomefile, "r");

    if (f==NULL)
        return FALSE;
    else
    {
        vs->num = 0;

        while (fscanf(f, "%d %d %d %f",
                    &v.tipo, &v.taglia, &v.quant, &v.prezzo)!=EOF)
        {
            vs->v[vs->num] = v;
            vs->num++;
        }
        fclose(f);
        return TRUE;
    }
}

/*-----*/
/* 2. Definizione funzione estrai() */
/* Estrae i vestiti di una determinata taglia */
/*-----*/

void estrai(int taglia, vestiti vs, vestiti *es)
{
    es->num = 0;

```

```

    for (int i=0; i<vs.num; i++)
    {
        if (vs.v[i].taglia==taglia)
        {
            es->v[es->num] = vs.v[i];
            es->num++;
        }
    }
}

/* ----- */
/* 3. Definizione funzione quant() */
/* Restituisce la quantità totale di vestiti di una data taglia */
/* ----- */

int quant(vestiti vs)
{
    int q = 0;

    for (int i=0; i<vs.num; i++)
        q+=vs.v[i].quant;

    return q;
}

/* ----- */
/* 3. Definizione funzione prezzo() */
/* Restituisce il prezzo totale dei vestiti di una data taglia */
/* ----- */

float prezzo(vestiti vs)
{
    float p = 0.0;

    for (int i=0; i<vs.num; i++)
        p+=vs.v[i].quant*vs.v[i].prezzo;

    return p;
}

/* ----- */
/* 7. Definizione funzione scrivi() */
/* Scrive nel file filename i contatori degli accessi */
/* ----- */

int scrivi(char nomefile[], vestiti vs)
{
    FILE *f;
    vestiti es;
    int q;
    float p;

    f = fopen(nomefile, "w");

    if (f==NULL) return FALSE;
    else
    {
        for(int i=42; i<=54; i=i+2)

```

```

        {
            estrai(i,vs,&es);

            q = quant(es);
            p = prezzo(es);

            fprintf(f, "%d %d %f\n", i, q, p);
        }
    fclose(f);
    return TRUE;
}

/* ----- */
/* Definizione funzione errore() (non richiesta ma utile) */
/* Scrive a video un messaggio di errore e termina programma con */
/* valore di ritorno uguale al codice di errore */
/* ----- */

void errore(int n, char msg[])
{
    printf("%s\n",msg);
    exit(n);
}

/* ----- */
/* Programma: conteggio      Modulo: main.c */
/* ----- */

#include <stdio.h>
#include <stdlib.h>
#include "funzioni.h"

/* ----- */
/* Main(): avendo definito come funzioni tutte le azioni principali */
/* contiene solo la logica di alto livello. */
/* ----- */

int main()
{
    vestiti vs;

    if (leggi("VESTITI.TXT", &vs)==TRUE)
    {
        if (scrivi("TOTALI.TXT", vs)==FALSE)
            errore(2,"Impossibile scrivere il file TOTALI.TXT");
        exit(0);
    }
    else errore(1,"Impossibile leggere il file VESTITI.TXT");
}

```