**ESERCITAZIONE DEL 18/03/2020**

**Pubblicare le singole soluzioni su gitlab**

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, prezzo 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 \*/**

**#include <stdio.h>  
#include <stdlib.h>  
#include <stdbool.h>  
  
#define MAX 20  
#define NAMEFILE "vestiti.txt"  
#define TOTAL "TOTALI.txt"  
static char arrayType [][20] = {"camicia","pantalone", "maglia"} ;  
static int dim = 0;  
  
typedef struct vestito {  
 unsigned int tipo;  
 unsigned int taglia;  
 unsigned int quantita;  
 float prezzo;  
}VS;  
  
  
typedef struct Vestiti {  
 VS vestiti[MAX];  
 int count;  
}VS\_ALL;  
  
  
bool leggi(char \* nomeFile , VS\_ALL \* ves);  
  
void estrai(unsigned int taglia, VS\_ALL vestiti, VS\_ALL \* vestiti2);  
  
int quant(VS\_ALL vestiti);  
  
float prezzo(VS\_ALL vestiti);  
  
bool scrivi(char \* nomeFile , VS\_ALL vestiti);**

**/\* -------------------------------------------------------------------- \*/**

**/\* -------------------------------------------------------------------- \*/**

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

**#define MAX 20  
#define NAMEFILE "vestiti.txt"  
#define TOTAL "TOTALI.txt"  
static char arrayType [][20] = {"camicia","pantalone", "maglia"} ;  
static int dim = 0;**

**/\* -------------------------------------------------------------------- \*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 1. Definizione strutture \*/**

**typedef struct vestito {  
 unsigned int tipo;  
 unsigned int taglia;  
 unsigned int quantita;  
 float prezzo;  
}VS;  
  
  
typedef struct Vestiti {  
 VS vestiti[MAX];  
 int count;  
}VS\_ALL;**

**/\* -------------------------------------------------------------------- \*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 3. Dichiarazione funzioni \*/**

**bool leggi(char \* nomeFile , VS\_ALL \* ves);  
  
void estrai(unsigned int taglia, VS\_ALL vestiti, VS\_ALL \* vestiti2);  
  
int quant(VS\_ALL vestiti);  
  
float prezzo(VS\_ALL vestiti);  
  
bool scrivi(char \* nomeFile , VS\_ALL vestiti);**

**/\*----------------------------------------------------------------------\*/**

**/\* -------------------------------------------------------------------- \*/**

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

**/\* -------------------------------------------------------------------- \*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 1. Definizione funzione leggi() \*/**

**/\* Legge da nomefile i dati delle righe nella struttura vestiti \*/**

**bool leggi(char \* nomeFile , VS\_ALL \* ves) {  
 int i = 0;  
 FILE \* FP = fopen(nomeFile, "r");  
 if(!FP) {  
 printf("file non trovato :(\n");  
 return false;  
 }  
  
 (\*ves).count = 0;  
 while (! feof(FP) ){  
 fscanf(FP,"%d %d %d %f", &(\*ves).vestiti[i].tipo, &(\*ves).vestiti[i].taglia, &(\*ves).vestiti[i].quantita, &(\*ves).vestiti[i].prezzo);  
 (\*ves).count++;  
 }  
 fclose(FP);  
 return true;  
}**

**/\* -------------------------------------------------------------------- \*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 2. Definizione funzione estrai() \*/**

**/\* Estrae i vestiti di una determinata taglia \*/**

**void estrai(unsigned int taglia, VS\_ALL vestiti, VS\_ALL \* vestiti2){  
 (\*vestiti2).count = 0;  
 for (int i = 0; i < vestiti.count ; ++i) {  
 if(taglia == vestiti.vestiti[i].taglia){  
 (\*vestiti2).vestiti[(\*vestiti2).count].taglia = vestiti.vestiti[i].taglia;  
 (\*vestiti2).vestiti[(\*vestiti2).count].tipo = vestiti.vestiti[i].tipo;  
 (\*vestiti2).vestiti[(\*vestiti2).count].prezzo = vestiti.vestiti[i].prezzo;  
 (\*vestiti2).vestiti[(\*vestiti2).count].quantita = vestiti.vestiti[i].quantita;  
 (\*vestiti2).count++;  
 }  
 }  
}**

**/\* ---------------------------------------------------------------------\*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 3. Definizione funzione quant() \*/**

**/\* Restituisce la quantità totale di vestiti di una data taglia \*/**

**int quant(VS\_ALL vestiti) { return vestiti.count;}**

**/\* ---------------------------------------------------------------------\*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* 3. Definizione funzione prezzo() \*/**

**/\* Restituisce il prezzo totale dei vestiti di una data taglia \*/**

**float prezzo(VS\_ALL vestiti) {  
 float sum = 0;  
 for (int i = 0; i < vestiti.count ; ++i) {  
 sum += vestiti.vestiti[i].prezzo;  
 }  
 return sum;  
}**

**/\* ---------------------------------------------------------------------\*/**

**/\* ------------------------------------------------------------------- \*/**

**/\* 7. Definizione funzione scrivi() \*/**

**/\* Scrive nel file filename i contatori degli accessi \*/**

**bool scrivi(char \* nomeFile , VS\_ALL vestiti) {  
 FILE \*FP = fopen(nomeFile, "a");  
 if(!FP){ return false; }  
 VS\_ALL v;  
 int count;  
 float sum;  
 for (int i = 0; i < vestiti.count ; ++i) {  
 if(vestiti.vestiti[i].taglia >= 42 && vestiti.vestiti[i].taglia <= 54 ){  
 estrai(vestiti.vestiti[i].taglia , vestiti , &v);  
 count = quant(v);  
 sum = prezzo(v);  
 fprintf(FP , "%d\t %d\t %f\n" , vestiti.vestiti[i].taglia, count, sum);  
 }  
 }  
 fclose(FP);  
 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 \*/**

**/\* -------------------------------------------------------------------- \*/**

**/\* ------------------------------------------------------------------- \*/**

**/\* Main(): avendo definito come funzioni tutte le azioni principali \*/**

**/\* contiene solo la logica di alto livello. \*/**

**/\* --------------------------------------------------------------------\*/**

**int main() {  
 VS\_ALL v;  
 if ( leggi(NAMEFILE,&v) == TRUE)  
 {  
 if (scrivi(TOTAL, v) == FALSE)  
 errore(2,"Impossibile scrivere il file TOTALI.TXT");  
 exit(0);  
 }else {  
 errore(1,"Impossibile leggere il file VESTITI.TXT");  
}  
  
 return 0;  
}**