

```

1  ///HEADER
2  //Francesco Cucchi 4BI
3
4  #include <stdio.h>
5  #include <stdlib.h>
6  #include <string.h>
7  #include <float.h>
8
9  ///COSTANTI
10 #define CONID 6
11 #define CONNAME 16
12 #define LEN 128
13
14 ///STRUTTURA RECORD
15 typedef struct{
16     char id[CONID];        ///univoco
17     char name[CONNAME];
18     float price;           ///>0
19     char tipo;             ///S-A-B
20     int nuovo;             ///0-1
21 }Tgioco;
22
23 ///PROCEDURE ESERCIZIO
24 Tgioco* imp(Tgioco *v, int *n);
25 int ric(Tgioco *v, char *el, int num);
26 void stampaCat(Tgioco *v, int num);
27 void stampa(Tgioco *v);
28 void valEcTot(Tgioco *v, int n);
29 void stampaPerID(Tgioco *v, int n);
30 void modPricePerID(Tgioco *v, int n);
31 void espCat(Tgioco *v, int n);
32
33 ///INPUT + VARIE
34 int lgInt(int vmin, int vmax, char* mex);        //LEGGI INT
35 float lgFlt(float vmin, float vmax, char* mex);  //LEGGI FLOAT
36 char lgChar(char *mex);                          //LEGGI CHAR
37 void lgStr(char *s, char *mex);                  //LEGGI STRING
38 void menu();
39 void err();
40
41 ///FUNZIONI
42 ///-----
43 #include "header.h"
44
45 /*
46     Francesco Cucchi 4BI
47
48     1) Importa il catalogo dei videogiochi da file (controlla che i dati del
49     videogiochi siano corretti. In caso di
50     errori la riga riferita al videogiochi dovra' essere esportata in un file
51     chiamato log.csv);
52
53     DESCRIZIONE FILE .CSV:
54     Il file csv conterra' i dati di un singolo videogiochi per ogni riga. Ogni dato
55     sara' separato da un ";", nel seguente ordine:
56     id;nome;prezzo;tipo;nuovo/usato;
57
58     PSEUDOCODIFICA:
59     INIZIO
60         apri il file f (input)
61         apri il file log (output)
62         se il file esiste
63         allora
64             leggi la prima riga del file f in str
65             k = 0
66             mentre il file non e' finito

```

```

64         spezza str nei vari item
65         assegna il primo item a v[k].id
66         ricerca v[k].id all'interno del vettore v e ritorna la sua posizione
in pos o -1
67         se pos == -1
68         allora
69             assegna il 2,3,4 item a v[k]. name, price e tipo
70             se tipo!= "S" && "A" && "B"
71             allora
72                 scrivi str nel file di log
73             altrimenti
74                 assegna l'ultimo item a nuovo
75                 se nuovo != 0 != 1
76                 allora
77                     scrivi str nel file di log
78                 altrimenti
79                     k++
80                 fse
81             fse
82         altrimenti
83             scrivi str nel file log
84         fse
85         leggi la prossima riga del file in str
86     fciclo
87     se k == 0
88     allora
89         scrivi "nessun gioco importato"
90     fse
91     chiudi il file f
92     chiudi il file log
93 altrimenti
94     errore
95 fse
96 FINE
97 */
98
99 Tgioco* imp(Tgioco *v, int *n){ ///importazione dei dati da file
100     FILE *f = fopen("f.csv","r"); //apro il file da cui importare i dati
101     if(f != NULL){ //se esiste
102         size_t dim = sizeof(Tgioco); //dimensione dell'array da istanziare
103         if(v != NULL) //se l'importazione e' gia' stata effettuata
104             free(v); //libera la zona di memoria corrispondente
105         v = (Tgioco*)malloc(dim); //allocazione di memoria
106         FILE *log = fopen("log.csv","w"); //apertura / creazione del file di log
107         *n = 0; //azzeramento contatore
108         int pos; //eventuale posizione dell'elemento duplicato
109         char str[LEN], aus[LEN]; //riga letta dal file | copia ausiliaria
110         fgets(str,LEN,f); //lettura della prima riga del file
111         while(!feof(f)){ //ciclo fino a terminazione file
112             strcpy(aus,str); ///copia della riga letta dal file in una stringa
ausiliaria per evitare problemi nella stampa sul file di log
113             strcpy((v+*n)->id,strtok(str,";")); //assegnazione primo item
114             pos = ric(v,(v+*n)->id,*n); //ricerca dell'univocita' dell'id
115             if(pos == -1){ //se non esiste
116                 strcpy((v+*n)->name,strtok(NULL,";")); //assegnazione secondo item
117                 (v+*n)->price = atof(strtok(NULL,";")); //assegnazione terzo item
118                 if((v+*n)->price <= 0){ //controllo sul prezzo
119                     printf("Contenuto riga invalido, prezzo non positivo\n");
120                     fprintf(log,"%s",aus);
121                     //esportazione riga su file di log + motivazione
122                 }
123             }
124             else{
125                 strcpy(&((v+*n)->tipo),strtok(NULL,";")); //assegnazione quarto
item
126                 if((v+*n)->tipo != 'S' && (v+*n)->tipo != 'A' && (v+*n)->tipo !=
'B'){ //controllo sulla validita' del file

```

```

126         printf("Contenuto riga invalido, TIPOLOGIA errata\n");
127         fprintf(log,"%s",aus);
128         //esportazione riga su file di log + motivazione
129     }
130     else{
131         (v+*n)->nuovo = atoi(strtok(NULL,";")); //importazione
ultimo item
132         if((v+*n)->nuovo != 0 && (v+*n)->nuovo != 1){
133             printf("Contenuto riga invalido, valore USATO errato\n"
);
134             fprintf(log,"%s",aus);
135             //esportazione riga su file di log + motivazione
136         }
137         else{
138             printf("Elemento num %d importato con successo\n", *n);
//conferma
139             (*n)++; //incremento contatore elementi corretti
140             dim += sizeof(Tgioco); //aumento dimensione
141             v = (Tgioco*)realloc(v,dim); //riallocazione con nuova
dimensione
142         }
143     }
144 }
145 }
146 else{
147     printf("Contenuto riga invalido, ID duplicato\n");
148     fprintf(log,"%s",aus);
149     //esportazione riga su file di log + motivazione
150 }
151     fgets(str,LEN,f); //lettura riga successiva
152 }
153     fclose(f);
154     fclose(log);
155 }
156 else
157     err();
158
159     return v;
160 }
161
162 int ric(Tgioco *v, char *el, int num){
163     int pos = -1;
164     for(int k = 0; k<num && pos == -1; k++){
165         if(strcmp((v+k)->id,el) == 0)
166             pos = k;
167     }
168     return pos;
169 }
170
171 void stampaCat(Tgioco *v, int num){ ///stampa ogni elemento importato
172     if(v != NULL){
173         for(int k=0; k<num; k++)
174             stampa(v+k);
175     }
176     else
177         printf("Importa prima i dati\n");
178 }
179
180 void stampa(Tgioco *v){
181     printf("ID: %s\n",v->id);
182     printf("Nome: %s\n",v->name);
183     printf("Prezzo: %.2f\n",v->price);
184     printf("Classificazione: %c\n",v->tipo);
185     printf("Nuovo: %d\n",v->nuovo);
186 }
187

```

```

188
189 void valEcTot(Tgioco *v, int n){ ///stampa valore economico totale
190     if(v != NULL){
191         float tot = 0;
192         for(int k=0; k<n; k++){
193             tot += (v+k)->price;
194         }
195         printf("Valore economico totale: %.2f\n", tot);
196     }
197     else
198         printf("Importa prima i dati\n");
199 }
200
201 void stampaPerID(Tgioco *v, int n){ ///stampa videogioco per ID
202     if(v != NULL){
203         char ausID[CONID];
204         lgStr(ausID, "Inserire ID interessato: ");
205         int pos = ric(v, ausID, n);
206         if(pos != -1)
207             stampa(v+pos);
208         else
209             printf("Nessun videogioco con tale ID\n");
210     }
211     else
212         printf("Importa prima i dati\n");
213 }
214
215 void modPricePerID(Tgioco *v, int n){ ///dato ID modifica il prezzo
216     if(v != NULL){
217         char ausID[CONID];
218         lgStr(ausID, "Inserire ID interessato: ");
219         int pos = ric(v, ausID, n);
220         if(pos != -1){
221             (v+pos)->price = lgFlt(0.01, FLT_MAX, "Inserire nuovo prezzo: ");
222             printf("Modifica effettuata con successo\n");
223         }
224         else
225             printf("Nessun videogioco con tale ID\n");
226     }
227     else
228         printf("Importa prima i dati\n");
229 }
230
231 void espCat(Tgioco *v, int n){ ///esporta tutti i videogiochi di una data categoria
232     char cat = lgChar("Inserire categoria ('S', 'A', 'B'): ");
233     int expd = 0;
234     while(cat != 'S' && cat != 'A' && cat != 'B'){
235         err();
236         cat = lgChar("Inserire categoria ('S', 'A', 'B'): ");
237     }
238     FILE* exp = fopen("expCat.csv", "w");
239     for(int k = 0; k<n; k++){
240         if((v+k)->tipo == cat){
241             fprintf(exp, "%s;%s;%.2f;%c;%d;\n", (v+k)->id, (v+k)->name, (v+k)->price
242 , (v+k)->tipo, (v+k)->nuovo);
243             expd++;
244         }
245     }
246     fclose(exp);
247     printf("Sono stati esportati %d elementi\n", expd);
248 }
249 ///MAIN
250 ///-----
251 #include "header.h"
252

```

```

253  /*
254      Francesco Cucchi      4^BI
255
256      TESTO:
257      Un negozio di informatica commissiona un applicativo per la gestione del proprio
catalogo di videogiochi.
258      Ogni videogioco ha le seguenti caratteristiche:
259      - ID - Univoco
260      - Nome
261      - Costo
262      - Classificazione (S, A, B)
263      - Usato (0->usato, 1->nuovo)
264      Realizzare un menu che presenti le seguenti funzionalita':
265      1) Importa il catalogo dei videogiochi da file (controlla che i dati del
videogioco siano corretti. In caso di
266      errori la riga riferita al videogioco dovra' essere esportata in un file
chiamato log.csv);
267      2) Stampa catalogo;
268      3) Stampa il valore economico totale del catalogo;
269      4) Dato l'ID, stampa i dati videogioco;
270      5) Dato l'ID, modifica il costo del videogioco;
271      6) Data la classificazione, esporta in un file i videogiochi aventi quella
classificazione;
272      - VINCOLI/OBBLIGHI
273      ? Il progetto dovrà essere realizzato dividendo il codice in piu' file.
274      ? Per gli input si dovranno utilizzare le apposite funzioni realizzate.
275      ? L'array di record si deve istanziare a run time
276      ? obbligo di uso dei puntatori all'interno delle procedure e delle funzioni.
277      ? Non è possibile utilizzare parentesi quadre ([]).
278  */
279
280  int main()
281  {
282      Tgioco* giochi = NULL;
283      int* num = (int*)malloc(sizeof(int));
284      int sc;
285      do{
286          system("cls");
287          menu();
288          sc = lgInt(0,6,"Inserire funzione interessata: ");
289          switch(sc){
290              case 1:{
291                  giochi = imp(giochi,num);
292                  break;
293              }
294              case 2:{
295                  stampaCat(giochi,*num);
296                  break;
297              }
298              case 3:{
299                  valEcTot(giochi,*num);
300                  break;
301              }
302              case 4:{
303                  stampaPerID(giochi,*num);
304                  break;
305              }
306              case 5:{
307                  modPricePerID(giochi,*num);
308                  break;
309              }
310              case 6:{
311                  espCat(giochi,*num);
312                  break;
313              }
314              case 0: break;

```

```

315         }
316         system("pause");
317         system("cls");
318
319     }while(sc!=0);
320 }
321
322 ///INPUT
323 ///-----
324 #include "header.h"
325
326 //Francesco Cucchi 4BI
327
328 int lgInt(int vmin, int vmax, char* mex){
329     int x;
330     printf("%s", mex);
331     scanf("%d", &x);
332     while(x<vmin || x>vmax){
333         printf("%s", "Errore\n");
334         printf("%s", mex);
335         scanf("%d", &x);
336     }
337     return x;
338 }
339
340 float lgFlt(float vmin, float vmax, char* mex){
341     float x;
342     printf("%s", mex);
343     scanf("%f", &x);
344     while(x<vmin || x>vmax){
345         printf("%s", "Errore\n");
346         printf("%s", mex);
347         scanf("%f", &x);
348     }
349     return x;
350 }
351
352 char lgChar(char *mex){
353     char c;
354     printf("%s", mex);
355     fflush(stdin);
356     scanf("%c", &c);
357     while(c == '\0'){
358         err();
359         printf("%s", mex);
360         fflush(stdin);
361         scanf("%c", &c);
362     }
363     return c;
364 }
365
366 void lgStr(char *s, char *mex){
367     printf("%s", mex);
368     fflush(stdin);
369     gets(s);
370     while(strcmp(s, "") == 0){
371         err();
372         printf("%s", mex);
373         fflush(stdin);
374         gets(s);
375     }
376 }
377
378 void menu(){
379     printf("1) Importazione catalogo;\n");
380     printf("2) Stampa catalogo;\n");

```

```
381     printf("3) Valore economico totale;\n");
382     printf("4) Dato ID, stampa dati;\n");
383     printf("5) Dato ID, modifica costo;\n");
384     printf("6) Esporta i videogiochi con una data la classificazione;\n");
385     printf("0) Termina programma;\n");
386 }
387
388 void err(){
389     printf("Errore\n");
390 }
```