

```
//DAVIDE BENEDETTI 4BI
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <limits.h>
#include <stdbool.h>

#define NMAX 100
#define LENMAX 250
typedef struct {
    char id[NMAX];
    char nom[NMAX];
    float costo;
    char classi;
    int usato;
}giochi;

int leggiInt(char msg[],int vmin,int vmax);
float leggiFloat(char msg[],int vmin,int vmax);
char leggiChar(char *msg);
void leggiStr(char *msg,char *s);
void imp(giochi *Tgiochi);
void stampa(giochi *Tgiochi);
void stampaCat(giochi *Tgiochi, int n);
int contaRighe();
bool controllaRiga(giochi *Tgiochi,int n);
void valEcon(giochi *Tgiochi,int n);
void cercaID(giochi *Tgiochi,int n);
void modCosto(giochi *Tgiochi,int n);
void stampaF(giochi Tgiochi,int n);
void stampaClasse(giochi *Tgiochi,int n);
void inizializzaFile();
```

```
#include "header.h"
```

```
void leggiStr(char *msg,char *s){
    printf("%s\n",msg);
    do{
        gets(s);
        if(strcmp(s,"")==0)
            printf("errore,reinserire");
    }while(strcmp(s,"")==0);
}
```

```
int leggiInt(char msg[],int vmin,int vmax){
    int n;
    printf("%s\n",msg);
    do{
        scanf("%d",& n);
        if(n<vmin || n>vmax)
            printf("errore,reinserire");
    }while(n<vmin || n>vmax);
    fflush(stdin);
    return n;
}
```

```
float leggiFloat(char msg[],int vmin,int vmax){
    float n;
    printf("%s\n",msg);
    do{
        scanf("%f",& n);
        if(n<vmin || n>vmax)
            printf("errore,reinserire");
    }
```

```

    }while(n<vmin || n>vmax);
    return n;
}

char leggiChar(char *msg){
    char s;
    printf("%s\n",msg);
    do{
        scanf("%c",& s);
        if(s=='\0')
            printf("\nerrore,reinserire\n");
    }while(s=='\0');
    return s;
}

//file csv che fra un token e l'altro e composto da ; esso vede 5 token per riga e una riga per elemento
// 324d;franco;12.2;B;0;
// 3456;alban;18.5;B;0;

```

```

#include "header.h"
/*Esercizio 2
Un negozio di informatica commissiona un applicativo per la gestione del proprio catalogo di videogiochi.
Ogni videogioco ha le seguenti caratteristiche:
? ID i% Univoco
? Nome
? Costo
? Classificazione (S, A, B)
? Usato (0->usato, 1->nuovo)

```

Realizzare un menu che presenti le seguenti funzionalità:

- 1) Importa il catalogo dei videogiochi da file (controlla che i dati del videogioco siano corretti. In caso di errori la riga riferita al videogioco dovrà essere esportata in un file chiamato log.csv);
- 2) Stampa catalogo;
- 3) Stampa il valore economico totale del catalogo;
- 4) Dato l'ID, stampa i dati videogioco;
- 5) Dato l'ID, modifica il costo del videogioco;
- 6) Data la classificazione, esporta in un file i videogiochi aventi quella classificazione;

```

? VINCOLI/OBBLIGHI
? Il progetto dovrà essere realizzato dividendo il codice in più file.
? Per gli input si dovranno utilizzare le apposite funzioni realizzate.
? L'array di record si deve istanziare a run time
? obbligo di uso dei puntatori all'interno delle procedure e delle funzioni.
? Non è possibile utilizzare parentesi quadre ([]).
*/

```

```

int main()
{
    int m;
    int n=0;
    n=contaRighe();
    printf("%d\n",n);
    giochi *Tgiochi=(giochi*)malloc(sizeof(giochi)*n);
    do{
        m=leggiInt("inserisci:\n1.Importa il catalogo dei videogiochi da file\n2.Stampa catalogo\n3.Stampa il
valore economico totale del catalogo\n4.Dato l'ID, stampa i dati videogioco \n5.Dato l'ID, modifica il costo del
videogioco \n6.Data la classificazione, esporta in un file i videogiochi aventi quella
classificazione\n7.termina\n",1,7);
        switch(m){
            case 1:
                imp(Tgiochi);
                break;
            case 2:
                stampaCat(Tgiochi,n);
                break;

```

```

        case 3:
            valEcon(Tgiochi,n);
            break;
        case 4:
            cercaID(Tgiochi,n);
            break;
        case 5:
            modCosto(Tgiochi,n);
            break;
        case 6:
            stampaClasse(Tgiochi,n);
            break;
        case 7:
            printf("termina");
            break;
    }
}while(m!=7);
return 0;
}

```

```

#include "header.h"

```

```

/*
DAVIDE BENEDETTI 3BI

```

```

variab loc

```

```

riga      puntatore su contenitore della riga del file da leggere      stringa

```

```

n          numero elementi

```

```

inizio

```

```

    apri file.csv in lettura

```

```

    se file i;½ stato aperto

```

```

    allora

```

```

        leggi una riga

```

```

        finche il file non e finito

```

```

            dividi la riga fra i suoi token (campi del record (Tgiochi+n) id,nom,costo,classi,usato

```

```

            controlla se i dati sono corretti

```

```

            se i dati sono corretti

```

```

            allora

```

```

                n++

```

```

            leggi prossima riga

```

```

        fineciclo

```

```

    altrimenti

```

```

        scrivi file non aperto

```

```

    chiudi file

```

```

fine

```

```

*/

```

```

void imp(giochi *Tgiochi){

```

```

    FILE *fin;

```

```

    char riga[LENMAX];

```

```

    int n=0;

```

```

    bool corretto;

```

```

    fin=fopen("file.csv","r");

```

```

    if(fin!=NULL){

```

```

        fgets(riga,NMAX,fin);

```

```

        while(!feof(fin)){

```

```

            strcpy((Tgiochi+n)->id, strtok(riga, ";"));

```

```

            strcpy((Tgiochi+n)->nom, strtok(NULL, ";"));

```

```

            (Tgiochi+n)->costo=atof(strtok(NULL, ";"));

```

```

            (Tgiochi+n)->classi = strtok(NULL, ";")[0];

```

```

            (Tgiochi+n)->usato=atoi(strtok(NULL, ";"));

```

```

            corretto=controllaRiga(Tgiochi,n);

```

```

            if(corretto)

```

```

                n++;

```

```

            fgets(riga,NMAX,fin);

```

```

    }
} else {
    printf("file non esiste");
}
//parametri formali
// Tgiochi      contiene i dati dei videogiochi(id,nom,costo,classi,usato)      di tipo giochi

//stampa i campi del record Tgiochi  id,nom,costo,classi,usato
void stampa(giochi *Tgiochi){
    printf("%s \n", (Tgiochi)->id);
    printf("%s \n", (Tgiochi)->nom);
    printf("%.2f \n", (Tgiochi)->costo);
    printf("%c \n", (Tgiochi)->classi);
    printf("%i \n", (Tgiochi)->usato);
}

/*
parametri formali
Tgiochi      contiene i dati dei videogiochi(id,nom,costo,classi,usato)      giochi
n            numero elementi                                           intero
i            contatore                                                intero

inizio
    mentre(i<n)
        stampa dati record Tgiochi
        i++
    fciclo
fine

*/
void stampaCat(giochi *Tgiochi, int n){
    for(int i=0; i<n; i++){
        stampa(Tgiochi+i);
    }
}

/*
inizio
    apri file in lettura
    conta righe del file
fine

*/
int contaRighe(){
    int numRighe=0;
    char riga[LENMAX];
    FILE *fin=fopen("file.csv", "r");
    if(fin!=NULL){
        fgets(riga, LENMAX, fin);
        while(!feof(fin))
        {
            numRighe++;
            fgets(riga, LENMAX, fin);
        }
    }
    else
        printf("Il file non esiste \n");
    fclose(fin);
    return numRighe;
}

/*
parametri formali
Tgiochi      contiene i dati dei videogiochi(id,nom,costo,classi,usato)      giochi
n            numero elementi                                           intero

```

var locali

i	contatore	intero
corretto	restituisce 1 se dati corretti altrimenti 0	bool

```
inizio
    i=0
    mentre(i<n)
        controlla che dati del record siano corretti altrimenti corretto=false
        i++
    fciclo
fine
```

```
*/
bool controllaRiga(giochi *Tgiochi, int n){
    bool corretto=true;
    for(int i=0; i<n && corretto; i++){
        if(strcmp((Tgiochi+n)->id, (Tgiochi+i)->id)==0){
            corretto=false;
        }
        if((Tgiochi+n)->classi!='S' && (Tgiochi+n)->classi!='A' && (Tgiochi+n)->classi!='B')
            corretto=false;
        if((Tgiochi+n)->usato!=0 && (Tgiochi+n)->usato!=1)
            corretto=false;
    }
    return corretto;
}
```

```
/*
parametri formali
Tgiochi    contiene i dati dei videogiochi(id,nom,costo,classi,usato)    giochi
n          numero elementi                                              intero
```

var locali

i	contatore	intero
tot	costo totale di tutti i videogiochi	reale

```
inizio
    i=0
    mentre(i<n)
        calcola il costo totale dei videogiochi
        stampa il valore del costo totale dei videogiochi
    fciclo
fine
```

```
*/
void valEcon(giochi *Tgiochi, int n){
    float tot=0;
    for(int i=0; i<n; i++){
        tot=tot+(Tgiochi+i)->costo;
    }
    printf("%.2f\n", tot);
}
```

```
/*
parametri formali
Tgiochi    contiene i dati dei videogiochi(id,nom,costo,classi,usato)    giochi
n          numero elementi                                              intero
```

var locali

i	contatore	intero
trovato	variabile per feedback	bool
el	id da ricercare	stringa

inizio

```

i=0
leggi in el l'id da ricercare
mentre(i<n e non trovato)
    se(el=id)
        stampa dati del record (Tgiochi+i)
    i++
fciclo
fine

*/
void cercaID(giochi *Tgiochi,int n){
    bool trovato=false;
    char el[LENMAX];
    leggiStr("inserisci id da ricercare",el);
    for(int i=0;i<n && !trovato;i++){
        if(strcmp((Tgiochi+i)->id,el)==0){
            trovato=true;
            stampa(Tgiochi+i);
        }
    }
    if(!trovato)
        printf("elemento non trovato");
}

/*
parametri formali
Tgiochi    contiene i dati dei videogiochi(id,nom,costo,classi,usato)    giochi
n          numero elementi                                              intero

var locali
el          id da ricercare                                              stringa
i          contatore                                                    intero
trovato     variabile per feedback                                      bool

inizio
i=0
leggi in el l'id da ricercare
mentre(i<n e non trovato)
    se(el=id )
        leggi il nuovo costo
    i++
fciclo
se non trovato
    allora
        scrivi el non trovato
fine

*/
void modCosto(giochi *Tgiochi,int n){
    bool trovato=false;
    char el[LENMAX];
    leggiStr("inserisci id da ricercare",el);
    for(int i=0;i<n && !trovato;i++){
        if(strcmp((Tgiochi+i)->id,el)==0){
            trovato=true;
            (Tgiochi+i)->costo=leggiFloat("inserisci nuovo prezzo",0,1000);
        }
    }
    if(!trovato)
        printf("elemento non trovato");
}

/*
parametri formali
Tgiochi    contiene i dati dei videogiochi(id,nom,costo,classi,usato)    giochi
n          numero elementi                                              intero

```

```

var locali
el          id da ricercare          stringa
i           contatore                intero
    trovato    variabile per feedback    bool

inizio
    i=0
    leggi in el l'id da ricercare
    mentre(i<n e non trovato)
        se(el=classe del record Tgiochi )
            stampa nel file classe.csv (Tgiochi+i)
        i++
    fciclo
    se non trovato
        allora
            nessun el trovato
fine

*/
void stampaClasse(giochi *Tgiochi,int n){
    bool trovato=false;
    char el;
    el=leggiChar("inserisci classe");
    inizializzaFile();
    for(int i=0;i<n;i++){
        if((Tgiochi+i)->classi==el){
            trovato=true;
            stampaF(*(Tgiochi+i),n);
        }
    }
    if(!trovato)
        printf("nessun elemento trovato");
}

/*
parametri formali
Tgiochi    contiene i dati dei videogiochi(id,nom,costo,classi,usato)    giochi
n          numero elementi                                              intero

inizio
    i=0
    apri il file classe.csv in append e
    svrivi i dati del record nel file
    chiudi file
fine

*/
void stampaF(giochi Tgiochi,int n){
    FILE *fin;
    fin=fopen("classe.csv","a");
    fprintf(fin,"%s;",Tgiochi.id);
    fprintf(fin,"%s;",Tgiochi.nom);
    fprintf(fin,"%f;",Tgiochi.costo);
    fprintf(fin,"%c;",Tgiochi.classi);
    fprintf(fin,"%d;\n",Tgiochi.usato);
    fclose(fin);
}

/*
inizio
    apri file classe csv in scrittura
    chiudi file
fine

*/

```

```
void inizializzaFile(){  
    FILE *fin;  
    fin=fopen("classe.csv","w");  
    fclose(fin);  
}
```