

```

//Programma per leggere tutto il file 1 RIGA alla volta
//Butta fuori un film per volta

//LISTE CONCAATENATE

#include <stdio.h>

//AGGIUNTO
//define Numero_Film 100 //uso il #define così che se aumentano i numeri dei film
//posso incrementare l'array senza problemi

enum tipo_colonna {Titolo = 0 , Genere , Lead_Studio , Audience_score , Profitability ,
Rotten_Tomatoes , Worldwide_Gross , Year };

struct film{ //definisco il "progetto"

    char Titolo[64];
    char Genere[64];
    char Lead_Studio[64];
    char Audience_score[4]; // uso "char" perchè noi dobbiamo acquisire Caratteri dal
file (in) --> il file è costituito da caratteri
    char Profitability[64];
    char Rotten_Tomatoes[4];
    char Worldwide_Gross[64];
    char Year[5]; // 2004 + \0 (carattere fine stringa)

};

/*
struct entry{

    struct film one_movie;
    struct entry *next;

}; */

void print_movie ( struct film movie ); //Prototipo

void riempimento_film ( struct film *current_movie, int carattere, int colonna, int
posizione ) ; //Prototipo

int main (void){

    FILE *in; //Dichiarazione --> puntatore di tipo FILE --> punterà al
file in input (in)
    int c; //variabile per memorizzare ciascun carattere (1 alla volta)

    char movies_db[] = "movies.csv"; //..._db = DataBase --> lo scriviamo
per chiarezza

    int colonna = 0; //numero della colonna nel file csv

    struct film *current_movie; //inizializzazione --> Puntatore -->
utilizzo il progetto per creare un qualcosa che si chiama "current_movie"
//*current_movie = puntatore alla struct film --> punta ogni volta al posto
successivo nella struct film

```

```

    struct film lista_film[numero_film];           //array che contiene i film    -->
    capienza max 100
    int film_count = 0;           //conteggio film letti -->         tiene in memoria a che film
    siamo

    //Aprire file di lettura "r"

    if( (in = fopen(movies_db, "r") ) == NULL ){
        printf("Non posso aprire il file in lettura \n");
        return 1;
    }

    //Legge il file dall'inizio alla fine e lo stampa nel terminale

    int i = 0;           //tiene traccia della posizione all'interno del singolo campo / genere

    current_movie = &lista_film[0];           // current_movie prende l'indirizzo alla cella 0
    dell'array lista_film (& = indirizzo --> si usa nelle variabili)

    while( ( c = getc(in) ) != EOF ){
        if( c == ',' ){
            riempimento_film ( current_movie, '\0', colonna, i );           //mettiamo '\0'
            per concludere la stringa / colonna

            colonna++;
            i = 0;

        } else if ( c == '\n' ){

            //SALVO IL FILM CORRENTE ED INCREMENTO COUNT

            riempimento_film ( current_movie, '\0', Year, i );           //Year --> va a
            prendere solo quella con Year           //lista_film[film_count] = passo il film corrente della lista
            di filmm           //lo scriviamo perche DI DEFAULT ogni
            movies / riga termina con '\n           // imponiamo lo '\0' perchè va alla fine
            di OGNI stringa

            current_movie++;           //per ogni film che leggo --> vado al successivo
            film_count++;
            i = 0;
            colonna = 0;

            //ALTRO MODO PER SCRIVERE "current_movie++" :
            //current_movie = &lista_film[film_count];           //prendo l'indirizzo del
            film suocessivo

            //Togliere da QUA
            //print_movie(current_movie);           //chiamando la funzione inserendo come
            valore in ingresso "current_movie"

```

```

        printf("\n");
    } else {
        //printf("%c", c);    //il carattere che leggiamo e salviamo

        riempimento_film ( current_movie, c, colonna, i );    //chiamata
        funzione con valori in ingresso

        i++;    //leggo il carattere    -->    lo salvo    -->    incremento il
        posto in cui salvare il prossimo carattere
    }

}

//DEVO STAMPARE TUTTI I FILM FUORI DAL WILES

for (int j = 0; j < film_count; j++) {
    print_movie(lista_film[j]);
    printf("\n");
}

printf("\n");

fclose(in);

return 0;
}

void read_movie(FILE *in, struct film *da_riempire){    // struct film *da_riempire    (è
un puntatore alla struct) -->

}

void print_movie ( struct film movie ){    //riceve : "struct film current_movie" dal
main

    printf("Titolo: %s\n", movie.Titolo);
    printf("Genere: %s\n", movie.Genere);
    printf("Lead Studio: %s\n", movie.Lead_Studio);
    printf("Audience score %: %s\n", movie.Audience_score);
    printf("Profitability: %s\n", movie.Profidability);
    printf("Rotten Tomatoes %: %s\n", movie.Rotten_Tomatoes);
    printf("Worldwide Gross: %s\n", movie.Worldwide_Gross);
    printf("Year: %s\n", movie.Year);

}

void riempimento_film ( struct film *current_movie, int carattere, int colonna, int
posizione ) {

    switch( colonna ){    //uso "colonna" perchè devo scegliere in che parte della
    struct mettere il carattere
        // case "enum" : struct . attributo_struct (di interesse) [i] (posizione) = c
        (carattere da inserire in posizione i)

```

```
        case Titolo : current_movie -> Titolo[posizione] = carattere; break;
        case Genere : (*current_movie).Genere[posizione] = carattere; break;
        case Lead_Studio: current_movie -> Lead_Studio[posizione] = carattere;
break;        // uso la -> perc accedere agli attributi della struct con un PUNTATORE
(current_movie)
        case Audience_score: current_movie -> Audience_score[posizione] = carattere;
break;
        case Profitability: current_movie -> Profidability[posizione] = carattere; break;
        case Rotten_Tomatoes: current_movie -> Rotten_Tomatoes[posizione] = carattere;
break;
        case Worldwide_Gross: current_movie -> Worldwide_Gross[posizione] = carattere;
break;
        case Year: current_movie -> Year[posizione] = carattere; break;
    }
}
```