```c
#include <conio.h>
#include <stdio.h>
#include <string.h>

#define MAX_PELICULAS   100
#define DIAS            7
#define MAX_LONG_NOMBRE 60
#define TOP10           10

// Declaracion de variables globales
typedef struct {
    int ID,Views_dia[DIAS];
    float Punt_dia[DIAS],Views_Total,Punt_Total;
    char Nombre[MAX_LONG_NOMBRE];
}peliculas;

peliculas ListaPeliculas[MAX_PELICULAS];

// Lista de funciones
void Inicializar (void);    // Inicializa la lista de peliculas por archivos basuras
void OPCION1 (void);        // Ingresa a la opcion 1 y realiza el procesamiento de datos
void OPCION2 (void);        // Ingresa a la opcion 2 y realiza ...
void OPCION3 (void);        // ...
void OPCION4 (void);        // ...
void CargarArchivo (FILE *archivo,int num_archiv);                  // Esta funcion se encarga de cargar los archivo al registro
void RANKING (void);                                                // Se encarga de procesar la opcion 1
void CargaTotalViews (void);                                        // Carga el total del views en el registro
int BuscaID (int id);                                               // Busca la id en el registro
void CargarTop10 (float Top10_views[],int Top10_id[]);              // Carga el top10 de peliculas en dos arreglos
void OrdenarListaPeliculas (void);                                  // Ordena el registro segun el total de views de forma descendete
void MostrarTop10 (float Top10_views[],int Top10_id[]);             // Muestra el top10 de views y su id
void GRABAR_TOP10(FILE *top10,float Top10_views[],int Top10_id[]);  // Graba en un archivo top10.txt los datos correspondientes
void AccionesComunes(void);                                         // Llama a funciones que son comunes a mas de una opcion
float VALORACION_PONDERADA(int lugar);
void DIA_MAS(int indice, float Views_t_dias[], char eldia[][15]); // Muestra el dia y la cantidad de las mayores visualizaciones

// Funcion principal //
// ---------------------------------- //
int main ()
{
    FILE *archivo_pelicula,*archivo_datos7dias;
    int opcion, cantpelis;

    printf("Bievenidos al programa\n");
    // Inicializamos el registro
    Inicializar();
    // Abre el archivo y carga los datos al registro
    CargarArchivo(archivo_pelicula,1);      // Abre el archivo peliculas.txt
    CargarArchivo(archivo_datos7dias,2);    // Abre el archivo datos_7dias.txt
```

```
50      // Inicializamos las funciones que son comunes a varias opciones
51      AccionesComunes();
52
53      printf("\nMenu:\n1:Ranking y top10\n2:Valoracion semanal\n3:Filtro cantidad de
visualizaciones\n4:Pico visualizaciones\n0:Cerrar\n\nOpcion:");
54      scanf("%d",&opcion);
55      while (opcion < 0 || opcion > 4)
56      {
57          printf("Ingrese de nuevo\n");
58          printf("\nMenu:\n1:Ranking y top10\n2:Valoracion semanal\n3:Filtro cantidad
de visualizaciones\n4:Pico visualizaciones\n0:Cerrar\n\nOpcion:");
59          scanf("%d",&opcion);
60      }
61
62      while (opcion != 0)
63      {
64          switch (opcion)
65          {
66              case 1:
67              {
68                  OPCION1();
69              break;
70              }
71              case 2:
72              {
73                  OPCION2();
74              break;
75              }
76              case 3:
77             {
78                  int views,i;
79                  printf("Ingrese cantidad de visualizaciones: ");
80                  scanf("%d", &views);
81                  printf("\n");
82                  cantpelis=0;
83                  i=0;
84               while(views < 0 || views > 99999999999)
85                  {
86                      printf("Valor no admitido, ingrese otro nuevamente:\n");
87                      scanf(" %d");
88                  }
89
90                  while(i<MAX_PELICULAS)
91                  {
92                      while(views<ListaPeliculas[i].Views_Total)
93                      {
94                          peliculas aux;
95                          int long_nombre = strlen(ListaPeliculas[i].Nombre); // strlen
96                          aux = ListaPeliculas[i];
97                          for (int indice = long_nombre; indice < 43; indice++)
98                          {
99                              strcat(aux.Nombre," ");   //
100                         }
101
102
 printf("================================================================================
==================\n");
103                         printf("Pelicula:\t\t\t\t\t\tVisualizaciones:\n%s\t\t%.0f\n",
aux.Nombre, aux.Views_Total);
```

```
 printf("================================================================================
==================\n");
                        cantpelis=cantpelis+1;
                        i=i+1;
                    }
                    i = i + 1;
                }
                if(cantpelis==0)
                    printf("Ningun titulo fue visto tantas veces!\n\n");
              while(views < 0 || views > 99999999999)
                {
                    printf("Valor no admitido, ingrese otro nuevamente:\n");
                    scanf(" %d");
                }
            break;
            }
            case 4:
            {
                OPCION4();
            break;
            }
        }
        printf("\nMenu:\n1:Ranking y top10\n2:Valoracion semanal\n3:Filtro cantidad
de visualizaciones\n4:Pico visualizaciones\n0:Cerrar\n\nOpcion:");
        scanf("%d",&opcion);
        while (opcion < 0 || opcion > 4)
        {
            printf("Ingrese de nuevo\n");
            printf("\nMenu:\n1:Ranking y top10\n2:Valoracion semanal\n3:Filtro
cantidad de visualizaciones\n4:Pico visualizaciones\n0:Cerrar\n\nOpcion:");
            scanf("%d",&opcion);
        }
    }
    printf("\nGracias por usar el programa");

    getch();
    return 0;
}
// ------------------------------------ //

// Funciones de inicializacion //
// ------------------------------------ //
void Inicializar (void)
{
    for (int i = 0; i < MAX_PELICULAS; i++)
    {
        ListaPeliculas[i].ID = 0;
        ListaPeliculas[i].Punt_Total = 0;
        ListaPeliculas[i].Views_Total = 0;
        for (int j = 0; j < DIAS; j++)
        {
            ListaPeliculas[i].Punt_dia[j] = 0;
            ListaPeliculas[i].Views_dia[j] = 0;
        }
    }
    return;
}
void CargarArchivo (FILE *archivo,int num_archivo)
{
```

```
160     if (num_archivo == 1)
161     {
162         if ((archivo = fopen("peliculas.txt", "r")) == NULL)
163             printf ( "Error en la apertura. Es posible que el fichero no exista \n");
164         else
165         {
166             for (int i = 0;i < MAX_PELICULAS; i++)
167             {
168                 if (!feof(archivo))
169                 {
170                     fscanf (archivo, "%d%s\n", &ListaPeliculas[i].ID,
    ListaPeliculas[i].Nombre);
171                 }
172             }
173         }
174     }
175     else if (num_archivo == 2)
176     {
177         int id_aux,dia,views,lista;
178         float punt;
179         if ((archivo = fopen("datos_7dias.txt", "r")) == NULL)
180             printf ( "Error en la apertura. Es posible que el fichero no exista \n");
181         else
182         {
183             while (!feof(archivo))
184             {
185                 fscanf (archivo, "%d%d%d%f",&id_aux,&dia,&views,&punt);
186                 lista = BuscaID(id_aux);
187                 if (lista == MAX_PELICULAS+1)
188                     printf("La pelicula no existe");
189                 else
190                 {
191                     ListaPeliculas[lista].Views_dia[dia-1] = views;
192                     ListaPeliculas[lista].Punt_dia[dia-1] = punt;
193                 }
194             }
195         }
196     }
197     fclose(archivo);
198     return;
199 }
200 // ------------------------------------ //
201
202 // Funciones de menu //
203 // ------------------------------------ //
204 void OPCION1 (void)
205 {
206     RANKING();
207     return;
208 }
209 void OPCION2 (void)
210 {
211     //variables
212     int cont,id,lugar;
213     float ponderada;
214
215     //inicio
216     printf("Id de la pelicula a buscar valoracion:");
217     scanf("%i",&id);
218     cont=0;
```

```
    // Buscamos la id de la pelicula para ver si la ingresada existe y corresponde con las guardadas
    for(int i=0; i < MAX_PELICULAS; i++)
    {
        if(id == ListaPeliculas[i].ID)
        {
            cont=cont+1;
            lugar=i;
        }
    }
    if(cont == 1)
    {
        ponderada = VALORACION_PONDERADA(lugar);
        printf("\nLa valoracion semanal de la pelicula %i ,%s es %.2f\n\n",ListaPeliculas[lugar].ID,ListaPeliculas[lugar].Nombre,ponderada);
    }
    else printf("\nEl codigo de pelicula no existe\n\n");
    return;
}
void OPCION4 (void)
{
    int indice;
    float Views_t_dias[7], num_may;
    char eldia[7][15]= {"lunes","martes","miercoles","jueves","viernes","sabado","domingo"};

    for(int j = 0; j < DIAS; j++) //limpia el arreglo
    {
        Views_t_dias[j] = 0;
    }

    //Suma el total de views de cada pelicula correspondiente a cada dia
    for(int lista = 0;lista < MAX_PELICULAS;lista++)
    {
        for(int dia = 0; dia < DIAS; dia++)
        {
      Views_t_dias[dia] = Views_t_dias[dia] + ListaPeliculas[lista].Views_dia[dia];
        }
    }

    for(int dia = 0; dia < DIAS; dia++)
    {
      if(dia==0) num_may=0, indice=0;

    if(Views_t_dias[dia] > num_may) num_may=Views_t_dias[dia], indice=dia;
  }

  DIA_MAS(indice,Views_t_dias,eldia);
}
// ------------------------------------ //

// ------------------------------------ //
// Funciones relacionadas con la opcion 4 //
// ------------------------------------ //
void DIA_MAS(int indice, float Views_t_dias[], char eldia[][15])
{
  printf("\n\n");
  printf("El dia con mas visualizaciones fue el %s",eldia[indice]);
  printf(" con %.0f",Views_t_dias[indice]);
  printf(" visualizaciones");
```

```
  printf("\n\n");

  return;
}
// ------------------------------------ //

// ------------------------------------ //
// Funciones relacionadas con la opcion 1 //
// ------------------------------------ //
void RANKING (void)
{
    FILE *archivo_top10;
    float Top10_views[TOP10];
    int Top10_id[TOP10];

    CargarTop10(Top10_views,Top10_id);
    //Muestra por pantalla toda la lista
    MostrarTop10(Top10_views,Top10_id);
    //Graba todo a un archivo
    GRABAR_TOP10(archivo_top10,Top10_views,Top10_id);
    // Permite no cargar dos o mas veces el total de views en la lista de peliculas

    return;
}
int BuscaID(int id)
{
    int indice;

    // Comienza la busqueda de la id, recorriendo la lista de peliculas existentes
    for (int i = 0; i < MAX_PELICULAS; i++)
    {
        if (ListaPeliculas[i].ID == id)
        {
            indice = i;
            return indice;
        }
    }
    // En caso de no encontrar el id de la pelicula devuelve el indice 101
    // dando a entender que no existe
    return MAX_PELICULAS+1;
}
void CargarTop10(float Top10_views[],int Top10_id[])
{
    // Limpiamos el arreglo para borrar los archivos basuras //
    for (int top10 = 0; top10 < 10; top10++)
    {
        Top10_views[top10] = 0;
        Top10_id[top10] = 0;
    }
    // Ordenamos la lista de peliculas según su total de visualizaciones //
    // OrdenarListaPeliculas ();
    //Copia los 10 primeros elementos del arreglo en el top10
    for (int top10 = 0; top10 < 10; top10++)
    {
        Top10_views[top10] = ListaPeliculas[top10].Views_Total;
        Top10_id[top10] = ListaPeliculas[top10].ID;
    }

    return;
}
```

```
void OrdenarListaPeliculas (void)
{
    peliculas aux;
    int lista,j,lugar;

    // Ordenamiento por seleccion //
    for ( lista = 0; lista < MAX_PELICULAS-1; lista++)
    {
        aux = ListaPeliculas[lista];
        lugar = lista;

        for ( j = lista+1; j < MAX_PELICULAS; j++)
        {
            if (ListaPeliculas[j].Views_Total > aux.Views_Total)
            {
                aux = ListaPeliculas[j];
                lugar = j;
            }
        }
        // Intercambia los registros //
        ListaPeliculas[lugar] = ListaPeliculas[lista];
        ListaPeliculas[lista] = aux;
    }
    return;
}
void MostrarTop10 (float Top10_views[],int Top10_id[])
{
    printf("\n");
    for (int i = 0; i < MAX_PELICULAS; i++)
    {
        printf("Id:\tTotal Views:");
        printf("\n%d",ListaPeliculas[i].ID);
        printf("\t%.0f",ListaPeliculas[i].Views_Total);
        printf("\n\n");
    }
    // Muestra por pantalla los datos
    printf("\nID:\tViews:\t");
    for (int top10 = 0; top10 < 10; top10++)
    {
        printf("\n%d\t%.0f",Top10_id[top10],Top10_views[top10]);
    }
    printf("\n");
    return;
}
void GRABAR_TOP10(FILE *top10,float Top10_views[],int Top10_id[])
{
    // Abre el archivo como escritura
    if ((top10 = fopen("top10.txt", "w")) == NULL)
            printf ( "Error en la apertura. Es posible que el fichero no exista \n");
    // Comienza a escribir
    fprintf(top10,"ID:\tNombre:\t\t\t\t\t\tViews:\n");
    for (int i = 0; i < TOP10; i++)
    {
        peliculas aux;
        int long_nombre = strlen(ListaPeliculas[i].Nombre); // strlen
        aux = ListaPeliculas[i];
        for (int indice = long_nombre; indice < 43; indice++)   // 43 es la cantidad
de caracteres del nombre mas largo
        {
            strcat(aux.Nombre," ");
```

```
        }

        fprintf(top10,"%d\t%s\t%.0f\n",Top10_id[i],aux.Nombre,Top10_views[i]);
    }
    // Cierra el archivo top10.txt
    fclose(top10);
    return;
}
// ------------------------------------ //

// Funciones relacionadas con la opcion 2 //
// ------------------------------------ //
float VALORACION_PONDERADA(int lugar)
{
    // Variables
    int total;
    float ponderada,puntuacion;

    puntuacion = 0;
    // Recorremos la lista buscando
    for(int i = 0; i < DIAS; i++)
    {
        puntuacion = puntuacion + ( ListaPeliculas[lugar].Views_dia[i] * 1.0 *
ListaPeliculas[lugar].Punt_dia[i] );
    }
    if(ListaPeliculas[lugar].Views_Total != 0) ponderada = puntuacion /
ListaPeliculas[lugar].Views_Total;
    else    return 0;
    return(ponderada);
}
// ------------------------------------ //

// Funciones relacionadas con varias opciones //
// ------------------------------------ //
void AccionesComunes(void)
{
    CargaTotalViews();
    OrdenarListaPeliculas ();

    return;
}
void CargaTotalViews(void)
{
    int dia,lista;

    //Suma el total de views de cada pelicula
    for (lista = 0;lista < MAX_PELICULAS; lista++)
    {
        for ( dia = 0; dia < DIAS; dia++)
        {
            ListaPeliculas[lista].Views_Total = ListaPeliculas[lista].Views_dia[dia]
+ ListaPeliculas[lista].Views_Total;
        }
    }
    return;
}
// ------------------------------------ //
```