```
//*****************************************************************************
//
//  NOMBRE DEL ARCHIVO: g4.c
//  MÓDULO:
//  FECHA: 3/04/2001
//  AUTOR:
//  DESCRIPCIÓN: Posible solución al examen parcial del grupo 4
//
//*****************************************************************************

#define __G4_C__

// Librerias estandar
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

// Otras librerías

// Encabezado Módulo-Proyecto

// Encabezado del archivo
#include "..\inc\g4.h"

// Variables globales externas

// Encabezado globales internas


//*********************************************************************
// Nombre: main
// Variables globales:
// Variables de salida:
// Comentarios (parámetros):
//
//*********************************************************************
void main(void)
{
    char nombre1[LONG_CADENA];
    char nombre2[LONG_CADENA];
    char cadena[LONG_CADENA];
    alumno *alumnos;

    int res,n;

    fflush(stdin);  // Limpia la entrada de teclado
    printf("Introducir el nombre del fichero sin extension:\n");
    gets(cadena);
    fflush(stdin);

    // Concatena el directorio y la extensión
    strcpy(nombre1, DIRECTORIO);
    strcat(nombre1, cadena);
    strcat(nombre1, EXTENSION);

    n = GenerarFichero(nombre1);
    if (n == -1)
    {
        printf("\n Error generando fichero. Programa terminado.\n");
        exit(1);
    }

    // Se leen los datos del fichero en una estructura con la longitud
    alumnos = (alumno *)malloc(n*sizeof(alumno));

    res = LeerFichero(nombre1,alumnos);
    if (res != 0)
    {
        printf("\n Error leyendo fichero. Programa terminado.\n");
        exit(1);
    }

    // Se ordenan las ciudades de menor a mayor número de habitantes
    Burbuja(alumnos,n);


    fflush(stdin);  // Limpia la entrada de teclado
    printf("\nIntroducir el nombre del fichero ordenado:\n");
```

```
    gets(cadena);
    fflush(stdin);

    // Concatena el directorio y la extensión
    strcpy(nombre2, DIRECTORIO);
    strcat(nombre2, cadena);
    strcat(nombre2, EXTENSION);

    fflush(stdin);

    res = CrearFicheroOrdenado(nombre2,alumnos,n);
    if (res != 0)
    {
        printf("\n Error generando fichero ordenado. Programa terminado.\n");
        exit(1);
    }

    printf("\nPrograma finalizado con exito.\n");

}

//*********************************************************************
// Nombre: GenerarFichero
// Variables globales:
// Variables de salida:
// Comentarios (parámetros): Crea un fichero a partir de entradas
//                           del teclado
//
//*********************************************************************
int GenerarFichero(char *nombre)
{
    FILE *pFich;
    char cadena[LONG_CADENA];
    int cont=0;
    int res;

    // abre el fichero para escritura
    pFich = fopen(nombre, "w");
    if (pFich == NULL)
    {
        printf("Error abriendo el fichero\n");
        return -1;
    }

    // Lee cadenas por teclado hasta que se pulse quit


    printf("\nIntroducir los nombres de los alumnos (salir para terminar):\n");
    do
    {
        printf("Nombre: ");
        fflush(stdin); // Limpia la entrada de teclado
        gets(cadena);
        res = (strcmp(cadena,SALIR_MIN) && strcmp(cadena,SALIR_MAY));
        if (res != 0)
        {
            // Escribe en el fichero
            fprintf(pFich, "%s\n", cadena);
            cont++;
        }
    } while (res != 0);

    // Cierra el fichero
    fclose(pFich);

    return cont;
}


//*********************************************************************
// Nombre: LeerFichero
// Variables globales:
// Variables de salida:
// Comentarios (parámetros): Lee el fichero de los nombres y lo almacena
//                           en el vector de nombres
//*********************************************************************
int LeerFichero(char *nombre,alumno *alumnos)
{
```

```
    FILE *pFich;
    char cadena[LONG_CADENA];
    int cont=0,longitud;

    // abre el fichero para lectura
    pFich = fopen(nombre, "r");
    if (pFich == NULL)
    {
        printf("Error abriendo fichero\n");
        return -1;
    }

    // Lee nombre y longitud de cadena
    while (!feof(pFich))
    {
        fgets(cadena,LONG_CADENA,pFich);
        longitud = strlen(cadena) - 1;
        cadena[longitud] = '\0';
        strcpy(alumnos[cont].nombre, cadena);
        alumnos[cont].numero = strlen(cadena);
        cont++;
    }

    // Cierra el fichero
    fclose(pFich);

    return 0;
}


//*********************************************************************
//  Nombre Función: Burbuja
//  Variables globales:
//  Variables de salida:
//  Comentarios (parámetros): Ordena el vector v de dimensión n utilizando
//                            el método de la burbuja. Ordena de menor
//                            a mayor longitud de cadena
//*********************************************************************
void Burbuja(alumno v[], int n)
{
    alumno aux;
    int i,j;

    for (i=1; i<n; i++){
        for (j=n-1; j>=i; j--){
            if (v[j-1].numero > v[j].numero){
                aux = v[j-1];
                v[j-1] = v[j];
                v[j] = aux;
            }
        }

    }

}



//*********************************************************************
// Nombre: CrearFicheroOrdenado
// Variables globales:
// Variables de salida:
// Comentarios (parámetros): Crea un fichero de alumnos ordenado por
//                             longitud de cadenas
//
//*********************************************************************
int CrearFicheroOrdenado(char *nombre, alumno *alumnos, int n)
{
    FILE *pFich;
    int i;

    // abre el fichero para escritura
    pFich = fopen(nombre, "w");
    if (pFich == NULL)
    {
        printf("Error abriendo el fichero\n");
        return -1;
    }
```

```
    // Se guardan los datos ordenados
    for(i=0;i<n;i++) fprintf(pFich,"%s\n",alumnos[i].nombre);

    // Cierra el fichero
    fclose(pFich);

    return 0;
}
```