

**Programación para Sistemas. EXAMEN PRÁCTICO PROGRAMACIÓN C**  
**Curso 2020/2021 Enero 2021. Duración 1 hora y 30 minutos**

El código del programa solicitado se debe escribir con LETRA LEGIBLE y respetando la PLANTILLA que se suministra. No se podrá utilizar hojas adicionales, si se necesita hacer anotaciones en sucio se puede emplear la parte de atrás de las hojas del enunciado. Se entregaran las hojas de la plantilla rellenas con el nombre y apellidos.

Se pide un programa en lenguaje C que recibirá los siguientes argumentos de entrada:

*Argumento 1:* Nombre fichero 1

*Argumento 2:* Una cadena de caracteres que actuará como separador

*Argumento 3:* Nombre fichero 2

El programa mostrará por la salida estándar líneas compuestas por las líneas del fichero 1 (argumento 1) y del fichero 2 (argumento 3) unidas por el separador (argumento 2), siguiendo la siguiente estructura:

```
Línea1_fichero1separadorLinea1_fichero2  
Línea2_fichero1separadorLinea2_fichero2  
Línea3_fichero1separadorLinea3_fichero2  
...  
Línean_fichero1separadorLinean_fichero2
```

Es importante observar que hay que **eliminar el posible santo de línea** de las líneas leídas en el fichero 1. Se puede asumir que las líneas se componen de caracteres de texto ASCII y con un tamaño máximo de 2048 caracteres.

El programa está compuesto por tres ficheros *main.c*, *cargar\_f1.c*, *unir\_f1\_f2.c* y un archivo cabecera *ExEne21.h*. Sea, 'unir\_lineas' el ejecutable resultado de la compilación y, por ejemplo, los ficheros fich1.txt y fich2.txt, cuyos contenidos se muestran con el comando 'cat':

```
cat fich1.txt          cat fich2.txt
Uno                    One
Dos                    Two
Tres                   Three
```

Al realizar la llamada *./unir\_lineas fich1.txt : fich2.txt* la salida estándar deberá ser:

```
Uno:One
Dos:Two
Tres:Three
```

Y si la llamada fuera *./unir\_lineas fich1.txt \_union\_ fich2.txt* la salida estándar deberá ser:

```
Uno_union_One
Dos_union_Two
Tres_union_Three
```

**El programa deberá respetar obligatoriamente las siguientes restricciones:**

1. Sólo se podrán leer las líneas de cada fichero una sola vez.
2. Deberán almacenarse las líneas del primer fichero en memoria dinámica antes de empezar a leer las líneas del segundo fichero. No es necesario almacenar las líneas del segundo fichero.
3. Deberá liberarse toda la memoria dinámica solicitada antes de finalizar el programa.

**Si el número de líneas de los ficheros de entrada es distinto**, se deberá generar las líneas de salida hasta que se acaben las líneas de alguno de los 2 ficheros y a continuación mostrar un mensaje indicando que fichero tiene menos líneas. Por ejemplo, si tuviéramos dos ficheros como los que se muestran a continuación:

cat fich1.txt	cat fich2.txt
Uno	One
Dos	Two
	Three
	Four

Al realizar la siguiente llamada `./unir_lineas fich1.txt : fich2.txt` la salida estándar deberá ser:

```
Uno:One
Dos:Two
STOP: Fichero 1 tiene menos líneas
```

Se deben tratar adecuadamente, mostrar en salida de error el mensaje especificado y finalizar el programa devolviendo al sistema operativo el código correspondiente, los siguientes **casos de error**:

- Si se proporciona un número de argumentos incorrecto: “ERROR: número de argumentos incorrecto”. Código error 64.
- Si hay error de lectura de algún fichero: “ERROR: No se ha podido abrir el archivo NOMBRE\_FICHERO”. Código error 66.
- Si hay algún error relacionado con la memoria dinámica: “ERROR: No se pudo asignar memoria dinámica”. Código error 71.

**ESCRIBIR EN LA PLANTILLA EL CÓDIGO. DEBES ENTREGAR SOLO LAS HOJAS DE RESPUESTA. NO ESTE ENUNCIADO.**

- **Hoja #1:** Includes, variables globales y código función `main()` : **30 pts**
- **Hoja #2:** Includes y tipos en `ExEne21.h` y código de `cargar_f1()` : **10 + 30 pts**
- **Hoja #3:** código funciones `unir_f1_f2()` y `liberar()` : **30 pts**

```

/* HOJA 1 */

#include "ExEne21.h"

lista_lineas *lista=NULL;

int main(int argc, char const *argv[]){

    FILE *ficherol;
    FILE *fichero2;

/*
 * Primera parte: procesado de args
 */
    if(argc != 4){
        fprintf(stderr,"ERROR: Número de argumentos incorrecto\n");
        exit(64);
    }

    fichero1 = fopen(argv[1],"r");
    if(fichero1 == NULL){
        fprintf(stderr,"ERROR: No se ha podido abrir el archivo %s
\n",argv[1]);
        exit(74);
    }

    fichero2 = fopen(argv[3],"r");
    if(fichero2 == NULL){
        fprintf(stderr,"ERROR: No se ha podido abrir el archivo
%s \n",argv[3]);
        exit(74);
    }

/* Segunda parte: carga en memdin */
    lista = cargar_f1(fichero1);
    if(lista==NULL)
        exit(66);

/* Tercera parte: unir líneas fich1 y fich2 */
    unir_f1_f2(fichero2, argv[2], lista);

/* Cuarta parte: salida ordenada */
    liberar(lista);
    fclose(fichero1);
    fclose(fichero2);

    exit(0);
}

```

```

/* HOJA 2 */
/* ExEne21.h */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct lineas {
    char *linea;
    struct lineas *siguiente;
} lista_lineas;

lista_lineas * cargar_f1(FILE *f);
void unir_f1_f2(FILE *f, const char *sep, lista_lineas *lptr);
void liberar(lista_lineas *lptr);

/** Cargar líneas del primer fichero en memoria dinámica
 * Se valorará específicamente la claridad y estructura del código
 */
lista_lineas * cargar_f1(FILE *f) {
    lista_lineas *head=NULL,*aux=NULL,*nuevo=NULL;
    char line[2049];
    int len;

    while(fgets(line,2049,f)){
        len=strlen(line);
        if(line[len-1]=='\n') {
            line[len-1]='\0';
        }

        nuevo=(lista_lineas *) malloc (sizeof(lista_lineas));
        if (nuevo==NULL) {
            fprintf(stderr,"ERROR: No se pudo asignar memoria dinámica \n");
            return NULL;
        }
        nuevo->linea=strdup(line);
        if (nuevo->linea==NULL) {
            fprintf(stderr,"ERROR: No se pudo asignar memoria dinámica \n");
            return NULL;
        }
        nuevo->siguiente=NULL;

        if (head==NULL) {
            head=nuevo;
            aux=head;
        } else {
            aux->siguiente=nuevo;
            aux=aux->siguiente;
        }
    }
    return head;
}

```

```

/* HOJA 3*/

/**
 * Unir líneas del segundo fichero e imprimir
 */
void unir_f1_f2(FILE *f, const char *sep, lista_lineas *lptr) {
    char line[2049];
    lista_lineas *aux;

    aux=lptr;

    while(fgets(line,2049,f) && aux!=NULL){

        fprintf(stdout,"%s%s%s",aux->linea,sep,line);
        aux=aux->siguiente;
    }

    if (feof(f) && aux!=NULL) {
        fprintf(stderr,"STOP: Fichero 2 tiene menos lineas \n");
    }

    if (!feof(f) && aux==NULL) {
        fprintf(stderr,"STOP: Fichero 1 tiene menos lineas \n");
    }
}

/**
 * Liberar memoria dinámica
 */
void liberar(lista_lineas *lptr) {
    lista_lineas *aux=lptr, *aux2;

    while(aux!=NULL){
        aux2=aux;
        aux=aux->siguiente;
        free(aux2->linea);
        free(aux2);
    }
}

```