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

El código del programa solicitado se debe escribir con LETRA LEGIBLE en las hojas de respuesta 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. Se entregaran por separado las hojas de respuesta rellenas con el nombre y apellidos con LETRA LEGIBLE. El enunciado no se entrega.

Se desea implementar un programa C que lea líneas de texto e imprima en la salida estándar las líneas siguiendo el siguiente criterio que se describe a continuación. Siendo P el número total de líneas con un número par de caracteres e I el número total de líneas con un número impar de caracteres, se imprimirá:

- Únicamente las líneas que contengan un número par de caracteres si P>I
- Únicamente las líneas que contengan un número impar de caracteres si P<I
- Ninguna línea si P=I o si no se ha leído ninguna línea.

Ejemplos de uso:

Si *fich_entrada1.txt* contiene las siguientes líneas, y suponiendo que todas las líneas acaban en el carácter nueva línea: *AAAA* *AAAA* *BBBBB* *BBBBB* *BBBBB* Entonces la llamada: ./exanen_julio_2021.exe fich_entrada1.txt mostrará: *BBBBB* *BBBBB* *BBBBB*	Si *fich_entrada2.txt* contiene las siguientes líneas, y suponiendo que todas las líneas acaban en el carácter nueva línea: *AAAA* *AA* *BB* Entonces la llamada: ./exanen_julio_2021.exe fich_entrada2.txt mostrará: *AAAA* *AA* *BB*

Las líneas de texto pueden provenir de un fichero o bien de la entrada estándar si se ejecuta el programa sin argumentos. Se puede asumir que las líneas se componen de caracteres de texto ASCII y con un tamaño máximo de 2048 caracteres. El carácter nueva línea (\n) se considera como otro carácter a efectos de contarlo.

El programa está compuesto por tres ficheros ***main.c***, ***cargar.c***, ***imprimir.c*** y un archivo cabecera ***exjul21.h*** que se proporciona al final de este enunciado. La solución planteada debe ser compatible con el fichero cabecera así como con la plantilla de cada hoja de respuesta. Hay que **respetar los prototipos de las funciones** y exjul21.h no puede ser modificado.

Internamente, el programa deberá almacenar las líneas leídas en una lista (`lista_lineas *lista`), **empleando memoria dinámica,** y deberá ir actualizando los contadores de líneas con un número de caracteres par (`contador_par`) y con un número de caracteres impar

(`contador_impar`). La función *imprimir()* imprimirá las líneas correspondientes y después **liberará toda la memoria dinámica solicitada**.

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 más de un argumento: “ERROR: número de argumentos incorrecto”. Código error 64.

- Si hay error de lectura del 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 puedo asignar memoria dinámica”. Código error 71.

NOTAS:
- Se recuerda que existe la función strlen para calcular la longitud de un string.
- A%B calcula el resto de la división entre A y B.

PUNTUACIÓN:
- ***Hoja #1:*** función *main() :* **30 puntos**
- ***Hoja #2:*** código de *cargar()* : **40 puntos**
- ***Hoja #3:*** código función *imprimir():* **30 puntos**

Fichero cabecera:

```
/* exjul21.h */


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

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

typedef struct contadores {
      lista_lineas *lista;
      int contador_par;
      int contador_impar;
} lista_contadores;

void cargar(FILE *f, lista_contadores *ptr);
void imprimir(lista_contadores *ptr);
```

Examen Programación Para Sistemas. 22 de junio de 2021 – HOJA DE RESPUESTA (1/3)

Apellidos, Nombre_____

DNI/NIE/Pasaporte _____ **Nº Matrícula** _____

Hoja #1 (main.c): función main() [30 ptos]:

```
#include "exjul21.h"
                                                    /*  Programa principal  */
int main(int argc, char const *argv[]) {

// Declaración de variables
        FILE *fichero1;
        lista_contadores *lc = NULL;

// Test nº de argumentos
        if(argc > 2){
                fprintf(stderr,"ERROR: NÃºmero de argumentos incorrecto\n");
                exit(64);
        }
// Definición lista_contadores invocando calloc
        lc = calloc (0, sizeof(lista_contadores));
        if (lc==NULL) {
                fprintf(stderr,"ERROR: No se pudo asignar memoria dinámica \n");
                exit(71);
        }
/*   Definición alternativa invocando malloc
        lc = malloc (sizeof(lista_contadores));
        if (lc==NULL) {
                fprintf(stderr,"ERROR: No se pudo asignar memoria dinámica \n");
                exit(71);
        }
        // Inicializaciones explícitas
        lc->contador_par = 0;
        lc->contador_impar = 0;
        lc->lista = NULL;

  Definición alternativa sin memoria dinámica
        lista_contadores slc;
        slc.contador_par = 0;
        slc.contador_impar = 0;
        slc.lista = NULL;
        lc = &slc;
*/
// Determinar si hay un fichero como argumento o es stdin
        if (argc==2) {
                fichero1 = fopen(argv[1],"r");
                if(fichero1 == NULL){
                        fprintf(stderr,"ERROR: No se ha podido abrir el archivo %s \n",argv[1]);
                        exit(74);
                }
                cargar(fichero1, lc);
                fclose(fichero1);
        }  else  // Sin argumento cargar desde stdin
                cargar(stdin, lc);

// Imprimir y liberar lista_contadores
        imprimir(lc);
// Solo en el caso de definiciÃ³n con memoria dinámica
        free(lc);
// Salir sin error
        exit(0);

}
```

Examen Programación Para Sistemas. 22 de junio de 2021 – HOJA DE RESPUESTA (2/3)

Apellidos, Nombre _____

DNI/NIE/Pasaporte _____ **Nº Matrícula** _____

Hoja #2 (cargar.c): código función cargar() [40 ptos]:

```
#include "exjul21.h"
/*
 * Carga líneas del fichero f y actualiza la lista y contadores en ptr.
 *
 */

void cargar(FILE *f, lista_contadores *ptr) {

 /* error_memoria decide si ha habido un error de memoria */
 int error_memoria = 0;

 /* linea almacena la última línea leída */
 char linea[2049];

 /* nuevo es un puntero a un nodo con la línea actual */
 lista_lineas *nuevo = NULL;

 /* íltimo es un puntero al último nodo de la cadena */
 lista_lineas *ultimo = NULL;

 while (!error_memoria && fgets(linea, 2049, f) != NULL) {

   nuevo = (lista_lineas *)malloc(sizeof(lista_lineas));
   error_memoria = nuevo == NULL;

   if (!error_memoria) {
     nuevo->linea = strdup(linea);
     error_memoria = nuevo->linea == NULL;

     if (!error_memoria) {
       if (strlen(linea) % 2 == 0)
         ptr->contador_par++;
       else
         ptr->contador_impar++;

       if (ultimo == NULL)
         /* ptr->lista apuntar  al primero */
         ptr->lista = nuevo;
       else
         ultimo->siguiente = nuevo;
       ultimo = nuevo;
     }
   }
 }

 if (error_memoria) {
   fprintf(stderr, "ERROR: No se pudo asignar memoria dinámica\n");
   exit(71);
 }
}
```

Examen Programación Para Sistemas. 22 de junio de 2021 – HOJA DE RESPUESTA (3/3)

Apellidos, Nombre _____

DNI/NIE/Pasaporte _____ **Nº Matrícula** _____

Hoja #3 (imprimir.c): código función imprimir() incluyendo liberación memoria [30 ptos]:

```
#include "exjul21.h"


void liberar(lista_lineas *lptr);

/**
 * 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);
	}
}


/* Imprimir las líneas que correspondan y liberar memoria dinámica */

 void imprimir (lista_contadores *ptr) {

lista_lineas *aux;

	aux=ptr->lista;

	if (ptr->contador_par != ptr->contador_impar)
		while(aux!=NULL){
			if (ptr->contador_par > ptr->contador_impar) {
				if (strlen(aux->linea) % 2 == 0)
					fprintf(stdout,"%s",aux->linea);
			}
			else {
				if (strlen(aux->linea) % 2 != 0)
					fprintf(stdout,"%s",aux->linea);
			}
			aux=aux->siguiente;
		}
	liberar(ptr->lista);
}
```