

Resolver los siguientes problemas en forma clara y legible en código ISO-C99.

1. Un docente de Algoritmos y Programación necesita planificar su próximo cuatrimestre. Para esto necesita contar cuántos feriados le van a interrumpir las clases. Necesita un TDA para esto.
 - a. Declarar la estructura que encapsula a este TDA `feriados_t`. Explicar qué representa cada miembro y documentar la invariante de representación.
 - b. Implementar la primitiva `size_t feriados_totales(const feriados_t *feriados, enum dia dia)`; que devuelva la cantidad de feriados que caen en el `dia` de la semana dado.
 - c. Implementar la primitiva `bool feriados_agregar(feriados_t *feriados, enum dia dia)`; que deje agregar un feriado más en ese `dia` de la semana.
2. El Ministerio del Interior decidió modernizarse y publicar el calendario de feriados en un archivo binario. El mismo es una sucesión de registros con el siguiente formato:

```
+-----+-----+-----+-----+
|  dia  |  mes  |    n  | motivo |
+-----+-----+-----+-----+
```

donde `dia`, `mes` y `n` son números de 1 byte, mientras que `motivo` es una cadena de caracteres de longitud `n` (no incluye el `'\0'`).

Se tiene una estructura `struct feriado { char dia, mes; char *motivo; }`;

Implementar una función `bool leer_feriado(FILE *f, struct feriado *feriado)`; que reciba un archivo `f` abierto en lectura binaria y una estructura `feriado` **no** inicializada y cargue los datos en la estructura. La función debe devolver `false` en caso de error.

3. Escribir un programa que se ejecute

```
$ ./feriados_de_mes mes archivo
```

donde `mes` es un número de mes y `archivo` es el nombre de un archivo **binario** de feriados, que imprima por `stdout` la información de cada uno de los feriados que peretenezcan a ese mes.

¡Suerte! :)