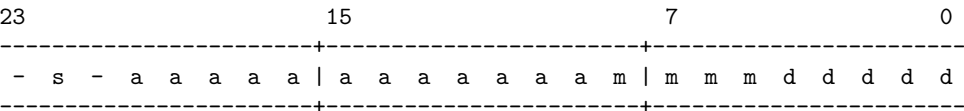


Resolver los siguientes problemas en forma clara y legible en código ISO-C99.

1. Una fecha puede representarse en un número de 24 bits según el siguiente formato

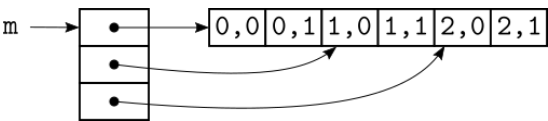


donde se utilizan 5 bits para el día, 4 bits para el mes, 12 bits para el año y un bit para el signo del año, donde si **s** vale 0 el año es positivo y si no negativo.

- a. Definir una estructura **tiempo** que tenga miembros para el día, mes y año.
- b. Definir el tipo **tiempo_t** en base a un entero no signado de 32 bits y el tipo **sttiempo_t** en base a la estructura del punto anterior.
- c. Implementar una función **void tiempo_a_sttiempo(tiempo_t t, sttiempo_t *st);** que convierta el tiempo en su representación entera **t** en la representación de la estructura y lo devuelva en **st**.

Nota: Los valores se representan igual en ambos formatos, por ejemplo, es indistinto si el mes se representa entre 0 y 11 o entre 1 y 12, la conversión es 1:1 del entero a la estructura.

2. Se propone representar a una matriz de enteros según, por ejemplo, la siguiente configuración para una matriz de 3x2:



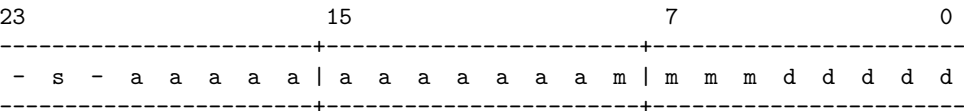
- a. Implementar la función **int **matriz_crear(size_t filas, size_t columnas);** que cree y devuelva una matriz de **filasxcolumnas** según el formato dado.
 - b. Implementar una función **void matriz_destruir(int **m, size_t filas, size_t columnas);** que reciba una matriz **m** de **filasxcolumnas** y libere su memoria asociada.
3. Implementar una función **char *leer_palabra();** que lea de **stdin** caracteres de a uno por vez hasta leer uno que no sea alfanumérico. Se debe devolver la cadena con los caracteres leídos (sin incluir el valor que no es alfanumérico).

Usar la función booleana **int isalnum(int c);** que indica si un valor es dígito o no. (Observación: **isalnum EOF** -> **false**)

¡Suerte! :)

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

1. Una fecha puede representarse en un número de 24 bits según el siguiente formato

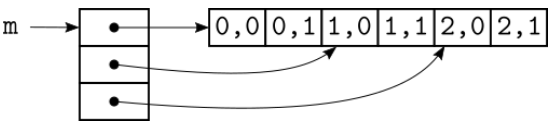


donde se utilizan 5 bits para el día, 4 bits para el mes, 12 bits para el año y un bit para el signo del año, donde si **s** vale 0 el año es positivo y si no negativo.

- a. Definir una estructura **tiempo** que tenga miembros para el día, mes y año.
- b. Definir el tipo **tiempo_t** en base a un entero no signado de 32 bits y el tipo **sttiempo_t** en base a la estructura del punto anterior.
- c. Implementar una función **void tiempo_a_sttiempo(tiempo_t t, sttiempo_t *st);** que convierta el tiempo en su representación entera **t** en la representación de la estructura y lo devuelva en **st**.

Nota: Los valores se representan igual en ambos formatos, por ejemplo, es indistinto si el mes se representa entre 0 y 11 o entre 1 y 12, la conversión es 1:1 del entero a la estructura.

2. Se propone representar a una matriz de enteros según, por ejemplo, la siguiente configuración para una matriz de 3x2:



- a. Implementar la función **int **matriz_crear(size_t filas, size_t columnas);** que cree y devuelva una matriz de **filasxcolumnas** según el formato dado.
 - b. Implementar una función **void matriz_destruir(int **m, size_t filas, size_t columnas);** que reciba una matriz **m** de **filasxcolumnas** y libere su memoria asociada.
3. Implementar una función **char *leer_palabra();** que lea de **stdin** caracteres de a uno por vez hasta leer uno que no sea alfanumérico. Se debe devolver la cadena con los caracteres leídos (sin incluir el valor que no es alfanumérico).

Usar la función booleana **int isalnum(int c);** que indica si un valor es dígito o no. (Observación: **isalnum EOF** -> **false**)

¡Suerte! :)