Algoritmos y Programación I (95.11) – Curso Kuhn – 2do parcialito – 13/05/2019

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

1. Se quiere almacenar los datos de alumnos. Un alumno tiene un "nombre" que es un texto de no más de `MAX_CADENA` elementos, un "padrón", que es un entero sin signo y un "promedio general", que es un número real.
   a. Declarar una estructura `struct alumno` que modele al alumno descripto.
   b. Definir el tipo `alumno_t` en base a la estructura anterior.
   c. Escribir una función `bool alumno_son_iguales(const alumno_t *a, const alumno_t *b);` que dados dos alumos `a` y `b` diga si son iguales. Un alumno es igual a otro si coinciden todos sus campos.
2. Teniendo definida una estructura `vector_t` la cual contiene un `float *v` y un `size_t n` y que modela a un vector dinámico `v` de `n` elementos se pide:
   a. Escribir una función `vector_t *vector_crear(const float vals[], size_t n);` que genere un `vector_t` de longitud `n` inicializado con los valores del vector `vals`.
   b. Escribir una función `void vector_destruir(vector_t *v);` que libere la memoria asociada a un `vector_t` dinámico.
3. Escribir una función `bool leer_enteros(int **pv, size_t *n);` que lea valores enteros de `stdin` y los almacene en un vector dinámico de enteros. Los parámetros `pv` y `n` son sólamente de salida y la función debe retornar a través de ellos el vector dinámico y su longitud. A su vez debe retornar por el nombre `true` si todo está bien o `false` en caso de error.

¡Suerte! :)

Algoritmos y Programación I (95.11) – Curso Kuhn – 2do parcialito – 13/05/2019

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

1. Se quiere almacenar los datos de alumnos. Un alumno tiene un "nombre" que es un texto de no más de `MAX_CADENA` elementos, un "padrón", que es un entero sin signo y un "promedio general", que es un número real.
   a. Declarar una estructura `struct alumno` que modele al alumno descripto.
   b. Definir el tipo `alumno_t` en base a la estructura anterior.
   c. Escribir una función `bool alumno_son_iguales(const alumno_t *a, const alumno_t *b);` que dados dos alumos `a` y `b` diga si son iguales. Un alumno es igual a otro si coinciden todos sus campos.
2. Teniendo definida una estructura `vector_t` la cual contiene un `float *v` y un `size_t n` y que modela a un vector dinámico `v` de `n` elementos se pide:
   a. Escribir una función `vector_t *vector_crear(const float vals[], size_t n);` que genere un `vector_t` de longitud `n` inicializado con los valores del vector `vals`.
   b. Escribir una función `void vector_destruir(vector_t *v);` que libere la memoria asociada a un `vector_t` dinámico.
3. Escribir una función `bool leer_enteros(int **pv, size_t *n);` que lea valores enteros de `stdin` y los almacene en un vector dinámico de enteros. Los parámetros `pv` y `n` son sólamente de salida y la función debe retornar a través de ellos el vector dinámico y su longitud. A su vez debe retornar por el nombre `true` si todo está bien o `false` en caso de error.

¡Suerte! :)