Algoritmos y Programación I (95.11) – Curso Essaya – 3er parcialito – 21/11/2019

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

1. Se quiere modelar el TDA votación, el cual contiene una determinada cantidad de cadenas que representan las opciones y los votos que tuvo cada opción. Se pide:

   a. Declarar la estructura que encapsula el TDA. Explicar qué representa cada miembro y documentar el invariante de representación.

   b. Implementar la primitiva `unsigned int votacion_votos_totales(const votacion_t *v);` que devuelva la cantidad de votos totales de todas las opciones.

   c. Implementar la primitiva `bool votacion_agregar_opcion(votacion_t *v, const char *opcion);` que agregue la opción a la lista de opciones.

2. Se tiene un archivo **binario** que contiene un `size_t` n seguido de n valores flotantes de doble precisión, según el siguiente esquema:

   ```
   +----+----+----+----+----+-   -+----+
   | n  | a1 | a2 | a3 | a4 | ... | an |
   +----+----+----+----+----+-   -+----+
   ```

   a. Escribir una función `bool escribir_doubles(const char *r, const double a[], size_t n);` que reciba una ruta `r` y un arreglo `a` de `n` doubles y los escriba en un archivo **binario** según el formato anterior.

   b. Escribir una función `double *leer_doubles(const char *r, size_t *n);` que reciba una ruta `r` a un archivo **binario** y devuelva por el nombre el arreglo de doubles contenido en él y en `n` la cantidad de elementos leídos.

3. Dado el formato (y las funciones) del ejercicio 2 escribir un programa que se ejecute:

   ```
   $ ./transformar entrada salida a b
   ```

   que cargue en memoria el arreglo de coordenadas contenido en el archivo **binario** `entrada` y aplique la transformación $ax + b$ a cada uno de sus elementos y lo guarde en el archivo **binario** `salida`.

   Se deben utilizar las funciones desarrolladas en el punto 2.

¡Suerte! :)

Algoritmos y Programación I (95.11) – Curso Essaya – 3er parcialito – 21/11/2019

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

1. Se quiere modelar el TDA votación, el cual contiene una determinada cantidad de cadenas que representan las opciones y los votos que tuvo cada opción. Se pide:

   a. Declarar la estructura que encapsula el TDA. Explicar qué representa cada miembro y documentar el invariante de representación.

   b. Implementar la primitiva `unsigned int votacion_votos_totales(const votacion_t *v);` que devuelva la cantidad de votos totales de todas las opciones.

   c. Implementar la primitiva `bool votacion_agregar_opcion(votacion_t *v, const char *opcion);` que agregue la opción a la lista de opciones.

2. Se tiene un archivo **binario** que contiene un `size_t` n seguido de n valores flotantes de doble precisión, según el siguiente esquema:

   ```
   +----+----+----+----+----+-   -+----+
   | n  | a1 | a2 | a3 | a4 | ... | an |
   +----+----+----+----+----+-   -+----+
   ```

   a. Escribir una función `bool escribir_doubles(const char *r, const double a[], size_t n);` que reciba una ruta `r` y un arreglo `a` de `n` doubles y los escriba en un archivo **binario** según el formato anterior.

   b. Escribir una función `double *leer_doubles(const char *r, size_t *n);` que reciba una ruta `r` a un archivo **binario** y devuelva por el nombre el arreglo de doubles contenido en él y en `n` la cantidad de elementos leídos.

3. Dado el formato (y las funciones) del ejercicio 2 escribir un programa que se ejecute:

   ```
   $ ./transformar entrada salida a b
   ```

   que cargue en memoria el arreglo de coordenadas contenido en el archivo **binario** `entrada` y aplique la transformación $ax + b$ a cada uno de sus elementos y lo guarde en el archivo **binario** `salida`.

   Se deben utilizar las funciones desarrolladas en el punto 2.

¡Suerte! :)