**Algoritmos y Programación I (95.11) – Curso Kuhn – 3er parcialito, 1er recuperatorio – 24/06/2019**

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

1. Se quiere modelar un TDA que represente una recta en $\mathbb{R}^2$ (como las implementadas en los EJ2 y EJ3). Se pide:

   a. Declarar la estructura `recta_t` que representa al TDA. Explicar qué modela cada uno de los miembros y sus valores posibles.

   b. Implementar la primitiva `bool recta_x_en_rango(const recta_t *r, float x);` que indique si la coordenada `x` está entre los extremos de la recta.

   c. Implementar la primitiva `bool recta_agregar_punto(recta_t *r, float x, float y);` que agrega un punto a la recta.

2. Se tiene un archivo **binario** que contiene un `size_t` $n$ seguido de $n$ pares de coordenadas flotantes, según el siguiente esquema:

   ```
   +----+----+----+----+----+-   -+----+----+
   | n  | x1 | y1 | x2 | y2 | ... | xn | yn |
   +----+----+----+----+----+-   -+----+----+
   ```

   Teniendo definido `typedef float coord_t[2];` se pide:

   a. Escribir una función `bool escribir_coordenadas(const char *r, coord_t *cs, size_t n);` que reciba una ruta `r` y un arreglo `cs` de `n` pares de coordenadas y los escriba en un archivo **binario** según el formato anterior.

   b. Escribir una función `coord_t *leer_coordenadas(const char *r, size_t *n);` que reciba una ruta `r` a un archivo **binario** y devuelva por nombre el arreglo de coordenadas 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

   ```
   $ ./rotar entrada salida radianes
   ```

   que cargue en memoria el arreglo de coordenadas contenido en el archivo **binario** `entrada`, rote cada uno de sus elementos por el ángulo `radianes` y lo guarde en el archivo **binario** `salida`.

   (Recordar las funciones de transformación $x' = x\cos\theta - y\sin\theta$, $y' = x\sin\theta + y\cos\theta$.)

¡Suerte! :)

**Algoritmos y Programación I (95.11) – Curso Kuhn – 3er parcialito, 1er recuperatorio – 24/06/2019**

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

1. Se quiere modelar un TDA que represente una recta en $\mathbb{R}^2$ (como las implementadas en los EJ2 y EJ3). Se pide:

   a. Declarar la estructura `recta_t` que representa al TDA. Explicar qué modela cada uno de los miembros y sus valores posibles.

   b. Implementar la primitiva `bool recta_x_en_rango(const recta_t *r, float x);` que indique si la coordenada `x` está entre los extremos de la recta.

   c. Implementar la primitiva `bool recta_agregar_punto(recta_t *r, float x, float y);` que agrega un punto a la recta.

2. Se tiene un archivo **binario** que contiene un `size_t` $n$ seguido de $n$ pares de coordenadas flotantes, según el siguiente esquema:

   ```
   +----+----+----+----+----+-   -+----+----+
   | n  | x1 | y1 | x2 | y2 | ... | xn | yn |
   +----+----+----+----+----+-   -+----+----+
   ```

   Teniendo definido `typedef float coord_t[2];` se pide:

   a. Escribir una función `bool escribir_coordenadas(const char *r, coord_t *cs, size_t n);` que reciba una ruta `r` y un arreglo `cs` de `n` pares de coordenadas y los escriba en un archivo **binario** según el formato anterior.

   b. Escribir una función `coord_t *leer_coordenadas(const char *r, size_t *n);` que reciba una ruta `r` a un archivo **binario** y devuelva por nombre el arreglo de coordenadas 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

   ```
   $ ./rotar entrada salida radianes
   ```

   que cargue en memoria el arreglo de coordenadas contenido en el archivo **binario** `entrada`, rote cada uno de sus elementos por el ángulo `radianes` y lo guarde en el archivo **binario** `salida`.

   (Recordar las funciones de transformación $x' = x\cos\theta - y\sin\theta$, $y' = x\sin\theta + y\cos\theta$.)

¡Suerte! :)