

## Algoritmos y Programación I (95.11) – Curso Santisi – 4<sup>to</sup> parcialito – 23/06/2023

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

1. Una *blockchain* es una lista enlazada donde los nuevos nodos se agregan al comienzo y contienen, además de un dato, un código de verificación que se compone de combinar el contenido del nodo con el código de verificación del nodo siguiente.

Tenemos una blockchain que almacena datos enteros con el nodo definido como `struct nodo { int dato; int verificacion; struct nodo *siguiente; };` y el TDA como `typedef struct { struct nodo *primero } blockchain_t` y donde se propone que la verificación sea la suma entre el dato y la verificación del nodo siguiente.

Implementar una primitiva `bool verificar_blockchain(const blockchain_t *b);` que devuelva `true` si la blockchain `b` es válida. Es decir, todos sus verificadores son correctos.

2. Agobiado ante la cantidad de mensajes en Discord un docente programó un bot que recopila los padrones del canal `#grupos-tp`, correspondientes a los alumnos que forman grupos, y pone cada uno de ellos ordenados en un arreglo.

Tiene además un listado de alumnos `struct alumno {char nombre[MAX_NOMBRE]; int padron; };` también ordenado por número de padrón.

Implementar una función

```
struct alumno *alumnos_antisociales(const struct alumno alumnos[], size_t nalumnos,
                                     const int padrones_grupos[], size_t npadrones, size_t *nantisociales);
```

que identifique **de forma eficiente** y devuelva un listado con los alumnos que no formaron grupo.

3. Se tiene un microprocesador que no puede ejecutar operaciones de división y producto. Implementar una función recursiva `unsigned int dividir(unsigned int a, unsigned int b);` que resuelva  $a/b$  utilizando sólo sumas o restas.

¡Suerte! :)