

## Parcialito Proyecto 3 - Tema X

Resolver los ejercicios listados abajo, en una hora (o menos). El código resultante **no** debe tener memory leaks **ni** accesos (read, write o free) inválidos a la memoria. La entrega es a través de Jaime.

### Ejercicio 1

Modificar el TAD bst tal que éste provea una operación nueva cuya signatura es:

```
unsigned int bst_leaf_count(bst_t bst)
```

Esta operación devuelve la cantidad de hojas que tiene el árbol dado como parámetro. Una hoja es aquel nodo cuyo árboles izquierdo y derecho son vacíos. Las PRE y POST de este método nuevo son las siguientes:

**pre** el árbol bst dado como parámetro es válido (y puede ser vacío).

**post** el resultado es exactamente la cantidad de hojas que tiene el árbol dado. Entre otras cosas, vale que:

```
bst_length(bst) = 0  $\Leftrightarrow$  result = 0  
result  $\leq$  bst_length(bst)
```

Para la implementación de este método **no** se deben hacer llamadas a otros métodos públicos del TAD bst, pero sí se pueden usar todos los métodos públicos de los otros TADs (es decir, del index, del data, del pair y de la lista).

### Ejercicio 2

Crear un archivo parcialito.c que provea una función main que haga lo siguiente:

- Crear un árbol vacío y agregarle los siguientes elementos (es **muy** importante poner los 0 del comienzo, cuando corresponde):
  - ("18", "dieciocho")
  - ("10", "diez")
  - ("50", "cincuenta")
  - ("05", "cinco")
  - ("15", "quince")
  - ("20", "veinte")
  - ("60", "sesenta")
  - ("02", "dos")
  - ("08", "ocho")
  - ("17", "diecisiete")
  - ("55", "cincuenta y cinco")

**Ayuda:** Para crear índices y datos usando cadenas de texto estáticas como las dadas arriba (es decir, strings que no usan memoria dinámica, por ende no hay que liberarlos), pueden hacer lo siguiente:

```
bst = bst_add(bst, index_from_string("perro"),  
             data_from_string("animal que ladra"));  
/* no hay que liberar ni el index ni el data, ni los strings estaticos */
```

- Llamar al método implementado en el punto 1 tal que se muestre por pantalla:
  - la cantidad de hojas del árbol vacío
  - la cantidad de hojas del árbol luego de cada valor que se le agregó