

# Proyecto 2

## primera parte

### Algoritmos y Estructuras de Datos II Laboratorio

8 de abril de 2010

1. Hacer el TAD lista implementado con listas enlazadas de punteros como se vió en el teórico del laboratorio. El header del TAD deberá ser:

```
#include "bool.h"

#include "alfa.h"

#ifndef LIST_H
#define LIST_H

typedef struct node *list;

list vacia(void);
list cons(alfa e, list l);
bool esvacia(list l);
alfa head(list l);
list tail(list l);
list destructor(list l);

#endif
```

Notar que hay que escribir los headers `bool.h` y `alfa.h`.

2. Con este TAD escribir una función en lenguaje C que calcule el segmento de suma mínima (Capítulo “Recursión final y programación imperativa” del libro Cálculo de Programas utilizado en Algoritmos I) utilizando sólo las operaciones sobre lista que brinda el mismo.

```
[[var ys : Int
  var a, b, r : Int
```

```

{R}
a, b, ys := 0, 0, xs
{P : G.(a, b).ys = G.(0, 0).xs}
do ys ≠ [ ] →
  a, b, ys := (head.ys + b) mín a, 0 mín(head.ys + b), tail.ys
od
{(a, b) = G.(0, 0).xs}
r := a
{Q : a = f.xs}
]]

```

**Nota:** Se debe hacer una interface a la función donde se carguen los datos por teclado. No debe haber memory leaks.

**Pregunta:** ¿Cuántas veces se recorre la lista?