

Proyecto 2

primera parte

Algoritmos y Estructuras de Datos II

Laboratorio

20 de abril de 2009

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 el programa que calcula 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  $ys \neq [] \rightarrow$ 
     $a, b, ys := (ys.0 + b) \text{ m\u00edn } a, 0 \text{ m\u00edn}(ys.0 + b), ys \downarrow 1$ 
od
 $\{(a, b) = G.(0, 0).xs\}$ 
 $r := a$ 
 $\{Q : a = f.xs\}$ 
 $\parallel$ 

```

Nota: No debe haber memory leaks.

Pregunta: ¿Cuántas veces se recorre la lista?