Ejercicio 1 .

TAD ConjuntoActado

obs elems : conj < Nat >
obs cota : Nat

proc ConjVacio(inn : Nat) : ConjuntoAcotado
asegura: res.elems = $\wedge$ res.cota = n

proc agregar(*inout* c : ConjAcotado, *in* e : Nat)
requiere: e <= c.cota
asegura: c.elems = old(c).elems $\cup$ e $\wedge$ c.cota = old(c).cota

proc pertenece(*in* c : ConjuntoAcotado, *in* e : Nat) : bool
asegura: res = true $\Leftrightarrow$ e $\in$ c.elems

modulo ConjAcotadoArr **implementa** ConjAcotada

var datos : Array < T >
var largo : int

pred invRep(c$'$: ConjAcotadoArr < T >)
$0 \leq$ largo $\leq$ c$'$.datos.lenght$\wedge$
$(\forall$i : Z$)(0 \leq$ i $<$ c$'$.largo $\wedge_{\mathrm{L}}$ #apariciones(c$'$.datos[i], subseq(c$'$.datos, 0, c$'$.largo)) = 1)

pred ABS(*in* c$'$: ConjAcotadoArr < T >, *in* c : conjAcotado < T >)
c.**cota** = c'.datos.lenght $\wedge$
$(\forall$e : T$)$(e $\in$ c.**elems** $\Leftrightarrow$ $(\exists$p : Z$)(0 \leq$ p $<$ c$'$.largo $\wedge_{\mathrm{L}}$ c$'$.datos[p] = e))

impl agregar(*inout* c : ConjAcotadoArr < T >, *in* e : T)
if c.pertence **then**
skip
else
c.datos[c.largo]:= e
c.largo = c.largo +1
end if

impl sacar(*inout* c : ConjAcotadoArr < T >, *in* e : T)
if pertenece(c,e) **then**
skip
else
i=0
while i < c.largo **and** c.datos[i] $\neq$ e **do**
i++
end while
if c.datos[i] = e **then**
c.datos[i]= c.datos[e.largo-1]
c.largo - -
end if
end if

impl pertence(*in* c : ConjAcotArr < T >, *in* e : T) : bool
i=0
while i < c.largo **do**
if c.datos[i] = e **then**
return True
else
skip
end if
end while
return False

Ejercicio 2 . Pila no acotada con Array

Idea: Imagino que tengo un array infinito, con **var inicio**, **var fin** que me indican donde empieza y termina mi pila, cuando apilo incremento mi var fin y reemplazo la posicion fin+1 por el nuevo elem , cuando desapilo decremento **var fin** (despues de la posicion fin no me importa que pasa).

Nota: para que la pila no sea acotada el array inicial deberia ser de largo infinito o bien lo suficientemente grande para cubrir la cantidad de veces que voy a apilar.

El ejercicio deberia ser identico a un tadAcotado pero con el array sin acotar.