2. Implementá el TAD Pila utilizando la siguiente representación:

**implement** Stack **of** T **where**

**type** Node **of** T = **tuple**
elem : T
next : **pointer to** (Node **of** T)
**end tuple**

**type** Stack **of** T = **pointer to** (Node **of** T)

```
implement Stack of T where

type Node of T = tuple
                   elem: T
                   next: pointer to (Node of T)
                 end tuple

type Stack of T = pointer to (Node of T)

constructors
    fun empty_stack() ret s: Stack of T
        s := null
    end fun

    proc push(in e: T, in/out s: Stack of T)
        var p: pointer to (Node of T)

        alloc(p)
        p→elem := e
        p→next := s
        s := p
    end proc

operations
    fun is_empty_stack(s: Stack of T) ret b: bool
        b := (s = null)
    end fun

    {- PRE: not is_empty_stack(s) -}
    fun top(s: Stack of T) ret e: T
        e := s→elem
    end fun

    {- PRE: not is_empty_stack(s) -}
    proc pop(in/out s: Stack of T)
        var p: pointer to (Node of T)

        p := s
        s := s→next
        free(p)
    end proc

end implement
```