



ALGORITMOS Y ESTRUCTURAS DE DATOS I

- Diciembre 2001 -

Dpto. Ing. Electrónica, Sistemas Informáticos y Automática
E.P.S. La Rábida (Universidad de Huelva)

Sección Informativa:

- El examen se compone de una sola parte de dos horas de duración, y consta de 3 ejercicios.
- Sólo se podrán realizar consultas durante los primeros 20 minutos, por lo que deberá leer atentamente ahora el examen completo.
- Se agotará la convocatoria transcurridos los primeros 20 minutos.
- Todos los ejercicios deberán ir correctamente comentados. No se dará por válido ningún ejercicio que cuente con tachones, falta de comentarios o letra ilegible.

Ejercicios:

1.- Escribir la función recursiva *SumaRecursiva* (*n*: entero): entero, que calcule la suma :

$$1+2+3+\dots+(n-1)+n$$

Por ejemplo *SumaRecursiva*(4) devolvería el valor 10 ya que

$$10 = 1+2+3+4$$

SumaRecursiva(5) devolvería el valor 15 ya que

$$15 = 1+2+3+4+5$$

2.- Sea la siguiente declaración de tipos:

tipo

lista = puntero a dato;

dato = registro

c: carácter;

siguiente : *lista*;

fin;

miLista = puntero a elemento;

elemento = registro

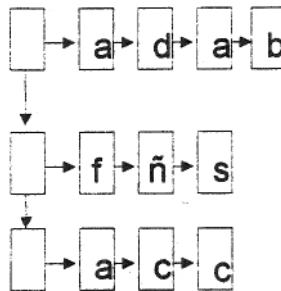
list : *lista*;

siguiente: *miLista*;

fin;

ftipo;

Gráficamente, podríamos representar la estructura especificada de la siguiente forma:



Basándonos en esta declaración de tipos:

a) Codificar la función

Cuenta (*L*: *miLista*, *c*: carácter): entero

la cual, dado el carácter *c*, cuenta cuántas incidencias del mismo aparecen en *L*. Por ejemplo, en el caso de la estructura representada en el gráfico, cuenta (*L*, 'a') devolvería el valor 3 ya que 'a' aparece 3 veces en toda la estructura.

b) Escribir la acción

EscribeFichero (*L*: *miLista*, *DATOS.DAT*)

la cual escribe el contenido de la estructura *L* en el mismo orden, en el fichero de tipo carácter *DATOS.DAT*, teniendo en cuenta que cada elemento del tipo *lista* está separada con un punto ('.'). Por ejemplo, el fichero equivalente a la estructura especificada gráficamente sería:

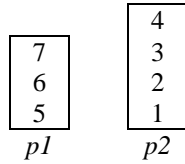
adab.fñs.acc. DATOS.DAT

El fichero *DATOS.DAT* está creado y cerrado. En caso de contener datos, deberán ser borrados.

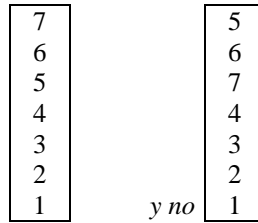
3. Implemente la acción

vuelcaCorrecto(p1: :pila,var p2:pila)

la cual vuelca el contenido de la pila p1 sobre la pila p2 en el orden en que se insertaron en la pila original. Por ejemplo, si tenemos las pilas:



Tras ejecutar *vuelcaCorrecto(p1,p2)* el valor de p2 deberá ser:



Para la realización de este ejercicio sólo pueden utilizarse los procedimientos y funciones propios del tipo pila.

Puntuación:

Ejercicio 1	2.5 puntos
Ejercicio 2 , apartado a)	2.5 puntos
apartado b)	2.5 puntos
Ejercicio 3	2.5 puntos