



Apellidos, Nombre

Grupo

Convocat.

Entregar esta hoja (con el Apellido, Nombre –en ese orden, y Grupo y número de la Convocatoria) delante de los folios con las respuestas

1 Responde las siguientes cuestiones brevemente (puedes emplear como máximo un folio para todas las respuestas.) 2

(a) Define los conceptos de *cohesión* y *acoplamiento*. .5

(b) ¿Para qué sirve y cómo funciona el *registro de activación* en las llamadas a subprogramas? Suponiendo definida la función recursiva **factorial(N)**, resuelve paso a paso la ejecución de la llamada **factorial(3)**, mostrando en cada paso el contenido de la pila de registros de activación. .5

(c) ¿Cuáles son las diferencias esenciales entre registro (o estructura) y objeto? .5

(d) ¿Qué condiciones tiene que verificar un algoritmo recursivo para que sea correcto y acabe? Pon un ejemplo de algoritmo recursivo y **razona** por qué es correcto y acaba. .5

_____ o O o _____

2 Implementa el TAD cola usando como representación el TAD lista posicional, tal como se definen en los siguientes interfaces de clase. 2

```
INTERFAZ CLASE CCola
TIPOS
  TBASE ...
METODOS
  Crear()
  Destruir()
  Encolar(E TBASE x)
  Desencolar()
  TBASE Frente()
  B EstaVacía()
FIN CCola
```

```
INTERFAZ CLASE CListaPos
TIPOS
  TBASE ...
METODOS
  Crear()
  Destruir()
  Insertar(E N pos, E TBASE x)
  Eliminar(E N pos)
  TBASE Consultar(E N pos)
  B EstaVacía()
FIN CListaPos
```

Observa que eres implementador de la clase cola y usuario de la clase lista posicional.

_____ o O o _____

3 Considera el siguiente interfaz de la clase CPila: 2.5

```
INTERFAZ CLASE CPila
TIPOS
  TBASE = ...
METODOS
  Crear()
  Destruir()
  Apilar(E TBASE elem)
  Desapilar()
  TBASE Cima()
  B EstaVacía()
FIN CPila
```

Una CPila puede representarse en memoria como una lista enlazada de *bloques*. Un *bloque* es similar a un nodo, excepto que en lugar de almacenar un solo elemento almacena varios elementos de la pila. Escribir la implementación correspondiente a la siguiente representación de pilas basada en bloques:

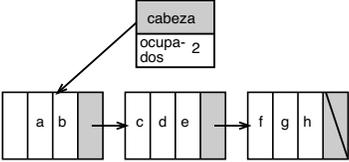
```

IMPLEMENTACION CLASE CPila
CONSTANTES
    TAM_BLOQUE= ...
TIPOS
    REGISTRO BLOQUE
        TipoElemento info[TAM_BLOQUE]
        BLOQUE* sig
    FIN REGISTRO
ATRIBUTOS
    BLOQUE* cabeza
    N ocupados
METODOS
    ...
FIN CPila

```

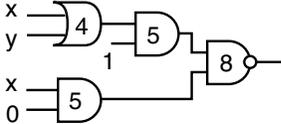
Observa que cada bloque almacena hasta

TAM_BLOQUE elementos de la pila. La pila se representa mediante dos atributos: un puntero **cabeza** que apunta al bloque en que se encuentra la cima de la pila y un natural **ocupados**, que indica cuántos elementos están ocupados en el bloque en que se encuentra la cima (los bloques por debajo de éste están completamente ocupados). La siguiente figura muestra una pila con TAM_BLOQUE = 3 que almacena 8 elementos.



_____ o O o _____

4 La siguiente figura representa un circuito combinacional cuyas puertas tienen un cierto retraso asociado que se representa por un número natural:



Un circuito combinacional se puede representar mediante un árbol binario cuyos nodos internos (no hoja) son puertas lógicas (*and*, *or* y *nand*) y cuyas hojas son bits (0 ó 1) o variables (letras minúsculas).

El tipo de los elementos del árbol binario se define como sigue:

```

ENUM TComp {puerta, bit, var}
ENUM TPuerta {and, or, nand}

REGISTRO TBASE
CASO TComp comp SEA
    puerta : TPuerta p
            N retraso
    bit : N b
    var : C v
FIN CASO
FIN REGISTRO

```

(a) Definir la función: ALGORITMO N MaxRetraso(E ArbolBin cto) que de-

vuelva el retraso máximo que se puede producir en el circuito representado por el árbol cto. Por ejemplo, para el circuito de la figura, el retraso máximo es 17.

(b) Definir el procedimiento ALGORITMO ExpAlg(E ArbolBin cto) que escriba la expresión algebraica equivalente al circuito. Toda subexpresión no atómica (es decir, que no sea un bit o una variable) debe aparecer entre paréntesis. Por ejemplo, para el circuito de la figura, la expresión algebraica equivalente es:

$$(((x \text{ or } y) \text{ and } 1) \text{ nand } (x \text{ and } 0))$$

(c) Definir el procedimiento: ALGORITMO VariablesDe(E ArbolBin cto, S ListaPos lv) que devuelva en la lista posicional lv la lista de variables del circuito. Las variables deben aparecer sólo una vez en lv. Por ejemplo, para el circuito de la figura la lista de variables es lv == [x,y].

3.5

Todos los apartados de este ejercicio deben resolverse empleando tanto el árbol binario como la lista posicional a nivel de usuario, suponiendo las interfaces estudiadas en clase, que deben ponerse (sólo las interfaces) antes de ser usadas.