



ALGORITMOS Y ESTRUCTURAS DE DATOS I

- Junio 2000 -

PARTE I

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

Sección Informativa:

- Tiempo disponible para la parte 1 : 1 hora y 30 minutos.
- No olvide poner sus apellidos, nombre, DNI, especialidad y grupo de teoría en todos los folios.

Teoría (2 puntos):

Conteste V (Verdadero) o F (Falso) a las siguientes cuestiones:

Puntuación:

Acierto: 0,25

Error: -0,25

Sin contestar: 0

1. Es correcto realizar la siguiente instrucción reservar(puntero a nodo).
2. La operación de liberar consiste en marcar la zona de memoria como disponible, para poder ser asignada a otra variable dinámica mediante la acción reservar, si fuera necesario.
3. El rango del tipo de dato puntero es el espacio de las direcciones de memoria más una dirección especial, la dirección nula.
4. La declaración:

```
tipo
    ref:puntero a nom;
    nom:entero;

ftipo
var
    r,rl:ref;

fvar
```

crea y asigna memoria a las variables apuntadas.

5. No se pueden realizar aperturas para lecturas sobre ficheros que físicamente que no existan.
6. Las aperturas para escritura que se realizan sobre ficheros que físicamente que no existen, producen la creación de dichos ficheros.
7. Si el identificador de fichero sobre el que se realice la operación de apertura para escritura existe, se borrará todo su contenido para dar entrada a los nuevos datos.
8. Dada la siguiente definición de tipo:

```
tipo
    punt = puntero a nodo;
    nodo = registro
    ...
    sig: punt;
    fregistro;
    lista = punt;

ftipo
funcion antes (p,q:punt; l:lista):booleano;
inicio
    retorna (p<q)

ffuncion
```

Devuelve *True* si el nodo apuntado por *p* está antes que el nodo apuntado por *q* en la lista.

Problema 1 (4 puntos): (enunciado muy confuso)

Se tienen N personas, identificadas por su nombre y su edad (entre 18 y 65 años) almacenadas en un fichero de acceso secuencial llamado "lista.dat". Se pide implementar un algoritmo que realice lo siguiente:

1. Acción que vuelque esta información en una lista circular en la que cada nodo almacena la información de una persona: nombre y edad, con la siguiente cabecera:

accion crea (var f:fich;var L:lista);

2. Acción que dado por teclado el nombre de una persona, devuelva un puntero al nodo de la lista que almacena su información, con la siguiente cabecera:

accion comienzo (var numero:entero; var pos: punt; L: lista);

(Siempre existe el elemento buscado).

3. Acción que dado el nodo de comienzo del paso anterior - apuntado por el puntero pos – seguirá el siguiente proceso:

- Eliminar la persona que está i posiciones a continuación del nodo de comienzo, en el sentido de las agujas del reloj, siendo i el resto de dividir su edad entre 10.
- Se tomará como nuevo valor de i el resto de dividir la edad de la última persona eliminada entre 10, el cuál, determinará la siguiente persona a eliminar.
- Este proceso se repite hasta que quede una única persona en la lista, el superviviente, y se mostrará su nombre por pantalla.

La cabecera de esta acción será :

accion elimina (numero:entero; pos:punt; var L:lista);

La declaración de tipos, variables globales y cuerpo del algoritmo se muestran a continuación:

algoritmo sobrevive;

tipo

nombre= tabla [1..30] de carácter;

persona= registro

 nomb: nombre;

 edad: entero;

fregistro

fich= fichero de persona;

punt= puntero a nodo;

nodo= registro

 info: persona;

 sig:punt;

fregistro;

lista= punt

ftipo

var

fich_per:fich;

lis_per:lista;

posicion:punt;

num:entero;

fvar

.....

inicio {Cuerpo del Algoritmo}

 crea (fich_per, lis_per);

 comienzo (num, posicion, lis_per);

 elimina (num, posicion, lis_per);

falgoritmo

NOTA: Se supone que están creadas las siguientes acciones y funciones:

accion leer_nom (var n: nombre);

{Almacena información de tipo nombre introducida por teclado}

accion escribir_nom (n:nombre);

{Visualiza por pantalla la información de tipo nombre}

funcion comparar (n:nombre; m:nombre):booleano;

{Devuelve verdad si n y m son iguales y falso en otro caso}



ALGORITMOS Y ESTRUCTURAS DE DATOS I

- Junio 2000 -

PARTE II

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

Sección Informativa:

- Tiempo disponible para la parte 2 : 1 hora y 30 minutos.
- No olvide poner sus apellidos, nombre, DNI, especialidad y grupo de teoría en todos los folios.

Cuestiones (1 punto):

Resuelva los siguientes apartados:

1.a. Averiguar qué hacen o qué devuelven las siguientes funciones recursivas:

1.a1.

funcion f1 (n:entero): entero;
var f:entero; fvar
inicio

si (n=0) \rightarrow f:= 0
□ (n<>0) \rightarrow

si (n mod 2 <> 0) \rightarrow f := f1(n-1) + n
□ (n mod 2 = 0) \rightarrow f:= f1(n-1) - n;

retorna (f)

ffuncion

1.a2.

funcion f2 (n:entero): entero;
var f:entero; fvar
inicio

si (n>100) \rightarrow f:= n-10
□ (n<= 100) \rightarrow f:= f2(f2(n+11));

retorna(f)

ffuncion

1.a3.

funcion f3 (x,y: entero): entero;
inicio

si (x<y) \rightarrow f3:= 0
□ (...) \rightarrow f3(x-y, y) +1;

ffuncion

1.b. Dada la función siguiente que determina el capital $C(n)$ a partir del capital inicial ci a un interés compuesto durante n años a un interés anual r . ¿Es correcta la siguiente implementación?

$$C(n) = Ci + (1 + r/100)^{**n}$$

funcion interesec (ci,n:entero;r:real):real;

inicio

 si $n=1 \rightarrow$ retorna $(ci * (1 + r/100)^{**n})$

 □ $n > 1 \rightarrow$ retorna $(\text{interesec}(ci, n-1, r) + (1 + r-1/100))$;

ffuncion

Problema 2 (3 puntos):

Se está realizando un programa para un ordenador el cual tendrá conectadas tres impresoras, cada una de las cuales idónea para un tipo de trabajo (la número 1 para borrador, la 2 para alta calidad y la 3 para color). El programa distribuirá los trabajos según su tipo y los encolará para cada impresora.

Todos los trabajos de impresión se encuentran en una lista *L* ordenada de modo que los primeros elementos de la lista son los trabajos de impresión menos prioritarios (los que deben imprimirse más tarde) y los últimos son los más prioritarios . (deben imprimirse antes). Por otro lado se va a disponer de tres colas *C1*, *C2* y *C3*, para almacenar los trabajos de cada impresora desde las que éstas se alimentarán. Los trabajos de impresión una vez entran en las colas ya no cambian su orden sino que se imprimen según van saliendo de su correspondiente cola.

Se pide implementar el subprograma Distribuidor tal que dada la lista *L* y las tres colas *C1*, *C2* y *C3*, cuando sea llamado realice la distribución de los trabajos de la lista a las colas manteniendo la prioridad temporal dentro de los trabajos de una misma impresora.

Datos:

Se le suministra un módulo de listas que exporta:

Los tipos:

- tipo_ele_L = registro

t_trab=(borrador,calidad,color); {Tipo trabajo}
trabajo: Datos; {Datos a imprimir }

fregistro

- Lista
- posición (para apuntar a nodos de la lista)

Los subprogramas:

- InicializaL(var L: Lista); {Prepara la lista L para trabajar con ella }
- InsertaL(ele:tipo ele_L; p:posicion; var L:Lista); { Inserta el elemento *ele* en la posición *p* de L }
- BorrarL(p:posicion; var L: Lista); {Elimina el elemento de la posición *p* de la lista L}
- VacíaL(L:Lista):booleana; {Función que devuelve verdadero si L está vacía}
- UltimoL(var p:posición; L: Lista); {Devuelve un puntero *p* al último elemento de L}
- Siguiete(p:posicion; L:Lista):posicion; {Función que devuelve un puntero al siguiente elemento de *p* en L},
- Ver(var ele:tipo_ele_L; p:posicion,vL:Lista); { Devuelve la información del nodo apuntado por *p* en la variable *ele* de la lista L }

Se le suministra un módulo de colas que exporta:

Los tipos:

- tipo_ele_C = Datos; { Información para imprimir }
- Cola

Los subprogramas:

- InicializaC(var C:Cola); { Prepara la cola C para trabajar con ella }
- MeterC(ele:tipo_ele_C; var C:Cola); { Introduce elemento en cola }
- SacarC(var ele:tipo_ele C; var C:Cola); { Extrae elemento de cola }
- VacíaC(C:Cola):booleano; { Consulta si la cola está vacía }



Universidad
de Huelva

ALGORITMOS Y ESTRUCTURAS DE DATOS I
SOLUCIÓN DE LA PARTE I (TEORÍA)
- Junio 2000 -

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

Cuestión 1	V
Cuestión 2	V
Cuestión 3	V
Cuestión 4	F
Cuestión 5	V
Cuestión 6	V
Cuestión 7	V