

Final

CADD

24-06-2025

Punto 2

- Ambos procedimientos son incorrectos,

El A borra correctamente el ultimo nodo de la lista, pero al pasar por referencia la lista y usar el puntero inicial para recorrerla y no un Auxiliar, se pierde el puntero inicial de la lista.

El B borra el ultimo nodo de la lista, pero no pasa la lista por referencia, por lo que el cambio no se ve afectado fuera del procedimiento.

Punto 3

program tres:

var c, d: integer;

procedure calcular (var a: integer; b: integer; var x: integer);

var d: integer;

begin

d := (b mod 4) + c;

b := (d mod 10) + d;

if (a + b > 25) then b := b + (a * 4);

12 + 12 < 25

else

x := (b + a) * 3; x = 24 * 3 = 72

c := (a - b) + c;

c = (12 - 12) + 3 = 3

writeln ('a: a, b: b, x: x, d: d');

end;

A = 12

B = 12

X = 72

D = 6

var a, b: integer;

begin

a := 4; b := 6; c := 3; d := 12;

calcular(d, c, a)

writeln ('a: a, b: b, c: c, d: d');

end

A = 72

B = 6

C = 3

D = 12

Mod. f. no global

sale 12

sale 72

entra 12

entra 3

entra 4

se usó esta variable

global

• el primer Write imprime $A=12$ $B=12$ $X=72$ $D=6$
el segundo imprime $A=72$ $B=6$ $C=3$ $D=12$

De inicio
Eing.

Punto 4

a) Verdadero. Una variable tipo simple puede aplicarse la estructura de control case. Esto no es siempre así porque la variable puede ser NO ordinal, en ese caso no se puede, pero en cualquier otro caso (sea que sea ordinal) sí se puede y el enunciado dice si se puede, por lo que es verdadero.

b) Falso: No siempre se puede recorrer un vector haciendo repeat until, porque este siempre se ejecuta al menos una vez, y en el caso de que la dimensión lógica sea 0, no tendría que ejecutarse nada.

c) Falso. Porque en la lista el ^{Acceso} ~~busqueda~~ es siempre Secuencial, en cambio en el Arreglo es directo, por lo que puede hacer una búsqueda dicotómica por ejemplo; que es más rápido que un Acceso secuencial.

Définir
Rangi

d) Memoria Estática y Dinámica FAISO

Constantes

- $\text{dimf} = 6 \text{ bytes}$

type

- $\text{Vector} = \text{dimf} \times \text{real} = 5 \times 8 = 40 \text{ bytes}$

- $\text{info} = 16 + 6 + 4 = 26 \text{ bytes}$

- $\text{lista} = 4 \text{ bytes}$

- $\text{nodo} = 26 + 4 = 30 \text{ bytes}$

Variables

$V = 40 \text{ bytes}$, $i = 6 \text{ bytes}$, $e = 26 \text{ bytes}$

$\text{OK} = 1 \text{ byte}$, $\text{monto} = 8 \text{ bytes}$, $l = 4 \text{ bytes}$, $\text{huel} = 4 \text{ bytes}$

$\text{MEMORIA ESTÁTICA} = \text{constantes} + \text{variables} = 6 + 89 = 95 \text{ bytes}$

- $\text{new}(\text{e.montos}) = 40 \text{ bytes}$

- $\text{new}(\text{huel}) = 30 \text{ bytes}$

$\text{MEMORIA DINÁMICA} = 40 + 30 = N 70 \text{ bytes}$

$\text{MEMORIA TOTAL} = M. \text{DINÁMICA} + M. \text{ESTÁTICA} = 95 \text{ bytes} + N 70 \text{ bytes}$

$95 \text{ bytes} + N 70 \text{ bytes}$

\neq

187 bytes

↓
FAISO

e) tiempo de ejecución FALSE

Deinde
Burg!

- Asignacion = $1ut$

- Asignacion = $1ut$

$$2N+3+25N$$

- While = $C(N+1) + N(\text{cuerpo}) = 3(N+1) + N(27) = 30N + 3ut$

$$C = \text{comparacion} + \text{conector} + \text{comparacion} = 2ut$$

$$N = \text{no se sabe}$$

$$\text{cuerpo} = it = 27ut$$

- $it = \text{condicion} + \text{cuerpo} = 2 + 25 = 27ut$

$$\text{condicion} = \text{comparacion} + \text{MOD} = 2ut$$

$$\text{cuerpo} = \text{for} + 3 \text{ Asignaciones} = 22 + 3 = 25ut$$

- $\text{for} = 3N + 2 + N(\text{cuerpo}) = 3 \cdot 5 + 2 + 5(1) = 22ut$

$$N = (5 - 1) + 1 = 5ut$$

$$\text{cuerpo} = \text{Asignacion} = 1ut$$

$$\text{tiempo total de ejecucion} = 2 \text{ Asignaciones} + \text{While}$$

$$= 2ut + 30N + 3ut = 30N + 5ut$$

$30N + 5ut$

≠

$$123ut$$

↓

FALSE