

Apellido y Nombre [REDACTED] De [REDACTED]



Por favor, no escribir sobre esta fotocopia, resolver los ejercicios en hojas aparte.

1. Un comercio dispone de una estructura de datos con información de las facturas realizadas durante el último mes (como máximo 1000). De cada factura se conoce el número de factura, código de sucursal (1..5) a la que pertenece y monto total. Se pide implementar un programa que lea de teclado un código de sucursal y elimine de la estructura todas las facturas pertenecientes al código de sucursal leído. Al finalizar debe informar el monto acumulado en todas las facturas eliminadas. La solución debe ser óptima en tiempo de ejecución.

2. Se quiere implementar un módulo que reciba un empleado y lo agregue a una estructura de empleados. Indique si las soluciones A y B lo resuelven de forma correcta. Justificar en ambos casos.

```
CONST CAPACIDAD=100;
type
  empleado = record
    cod: integer;
    nombre: String[30];
  end;

personal = record
  empleados: array [1..CAPACIDAD] of ^empleado;
  cantEmpleados: integer;
end;
```

A

```
procedure agregarEmpleado (p: personal; em: empleado);
begin
  if (p.cantEmpleados < CAPACIDAD) then begin
    p.cantEmpleados := p.cantEmpleados + 1;
    new(p.empleados[p.cantEmpleados]);
    p.empleados[p.cantEmpleados]^ := em;
  end;
end;
```

B

```
procedure agregarEmpleado (var p: personal; em: empleado);
begin
  if (p.cantEmpleados < CAPACIDAD) then begin
    p.cantEmpleados := p.cantEmpleados + 1;
    p.empleados[p.cantEmpleados]^ := em;
  end;
end;
```

¿Este programa indique qué imprime en cada sentencia write. Justifique su respuesta.

```
var tres;
var c, d: integer;

procedure calcular (var a: integer; var b: integer; c: integer);
var d: integer;
begin
  d := c MOD 2 + 15;
  a := c + a;
  if ((a + d) > 21) then b := a + d * 3
  else b := a + d;
  writeln ('Valor a: ', a, ' Valor b: ', b, ' Valor c: ', c, ' Valor d: ', d);
end;

var
  a, b: integer;
begin
  a := 4; b := 6; d := 12;
  calcular(b, c, a);
  writeln ('Valor a: ', a, ' Valor b: ', b, ' Valor c: ', c, ' Valor d: ', d);
end;
```

Indique Verdadero o Falso y justifique su respuesta en todos los casos:

- Si se conoce la cantidad máxima de elementos a almacenar, el vector es siempre la estructura de datos más adecuada para que la solución tenga el menor tiempo de ejecución.
- Corrección y eficiencia representan el mismo concepto.
- En un módulo que agrega un elemento al final de una lista, es correcto recibir el puntero inicial de la lista como un parámetro por valor.
- Las acciones dentro de una estructura de control **for** siempre se ejecutan al menos una vez.
- El **tiempo de ejecución** requerido por el programa "ejercicio4" no supera las 600 unidades de tiempo.
- La **memoria estática** requerida por el programa "ejercicio4" no supera los 80 bytes.
- La **memoria dinámica** requerida por el programa "ejercicio4" no supera los 1200 bytes.

```
program ejercicio4;
const
  MAX = 30; CANT_NOTAS = 4;
type
  rangoEstudiantes = 1..MAX;
  rangoNotas = 1..CANT_NOTAS;
  vector = array [rangoNotas] of ^real;
  estudiante = record
    ApyNom: string[30];
    legajo: string[10];
    notas: ^vector;
  end;
```

```
var
  e: estudiante; j: rangoNotas; nota: real; i:
  rangoEstudiantes;
begin
  for i:= 1 to MAX do begin
    read(e.ApyNom); read(e.legajo);
    new(e.notas);
    for j:= 1 to CANT_NOTAS do begin
      read(nota);
      new(e.notas[j]);
      e.notas[j]^ := nota;
    end;
```

Char	1 byte
Integer	6 bytes
Real	8 bytes
Boolean	1 byte
String	Longitud + 1 byte
Puntero	4 bytes