Final Conceptos de Algoritmos Datos y Programas — TEMA 1 — 10/10/2023

Apellido y Nombre: [redacted]

1. Un comercio dispone de una estructura de datos con las facturas realizadas durante agosto de 2023. De cada factura se conoce el número de factura, código de cliente, código de sucursal (1..5) y monto total. Se pide implementar un programa que lea un código de sucursal e invoque a un módulo que reciba dicho código y elimine las facturas correspondientes al código de sucursal recibida. Además debe retornar la cantidad de facturas eliminadas.

2.- Dada la siguiente declaración y los procesos A y B, indique para cada uno de ellos si elimina de forma correcta el primer elemento. **Justifique su respuesta**.

```
type
    numeros = array [1..100] of ^real;
    vector = record
      elem: numeros;
      diml: integer;
    end;
```

A

```
procedure eliminar1 (var v: vector);
var i: integer;
begin
  for i := 1 to (v.diml-1) do
    v.elem[i] := v.elem[i+1];
  if (v.diml > 0) then begin
    dispose(v.elem[i]);
    v.diml := v.diml - 1;
  end;
end;
```

B

```
procedure eliminar2 (var v: vector);
var i: integer;
begin
  for i := 1 to (v.diml-1) do
    v.elem[i]^ := v.elem[i+1]^;
  if (v.diml > 0) then begin
    dispose(v.elem[diml]);
    v.diml := v.diml - 1;
  end;
end;
```

3.- Calcule e indique la cantidad de **memoria estática y dinámica** que utiliza el siguiente programa. **Mostrar los valores intermedios para llegar al resultado y justificar.**

```
program ejercicio3;
const dimF = 120;
type
  cadena = string[45]; rango = 0..100;
  estudiante = record
    ape_nom: cadena; promedio: real; leg: integer;
  end;
  vector = array [0..dimF] of ^estudiante;
var
  v: vector; legajo: real; i, j: rango;
begin
i)for i:= 0 to 100 do begin
    new(v[i]); read(legajo); v[i]^.leg := leg;
  end;
ii)j:= 20;
iii)while(j > 0) and (v[j]^.leg <> 1234) do begin
    dispose(v[j]); j := j - 2;
  end;
end.
```

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

4.- Calcule el **tiempo de ejecución** del programa del punto 3. **Mostrar los valores intermedios para llegar al resultado y justificar.**

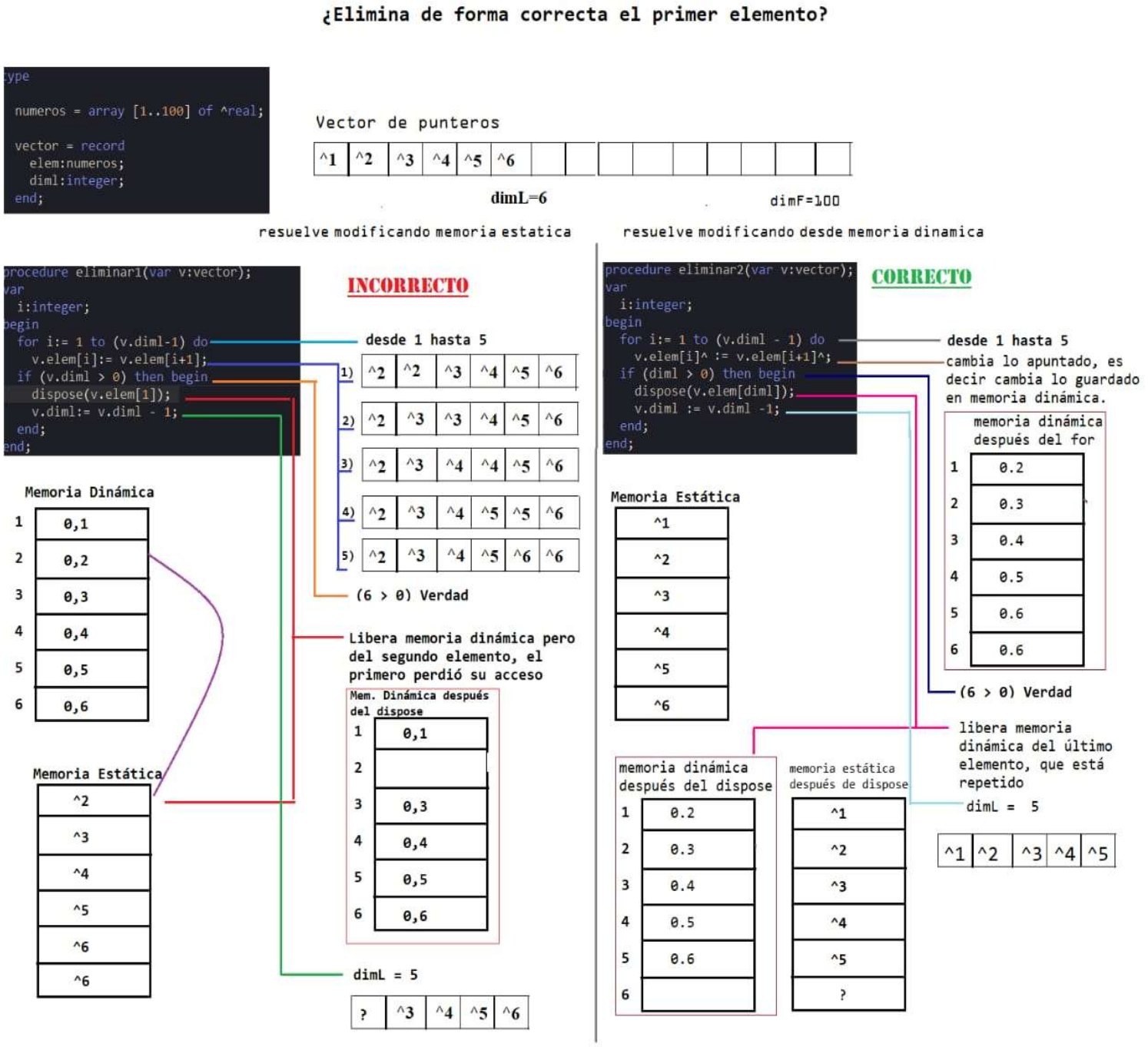
5.- Indique **Verdadero** o **Falso**. **Justifique** en todos los casos:

a. Un módulo función no puede retornar un tipo de dato puntero a un arreglo.
b. Si p es una variable de tipo puntero, entonces no es válida la siguiente instrucción: **readln**(p^).
c. Si un programa utiliza variables globales entonces no puede contener módulos que utilicen parámetros.
d. Un módulo con 5 líneas de código es más eficiente en tiempo de ejecución que un programa con 100 líneas de código.
e. Un tipo de dato registro puede ser homogéneo si todos sus campos son del mismo tipo de dato.
f. Nunca es posible acceder al nodo de la posición N en una estructura del tipo lista.

1.

```
program final2023;
type

    factura = record
        numero: integer;
        cliente: integer;
        sucursal: integer;
        monto: real;
    end;

    listaFacturas = ^nodo;

    nodo = record
        dato:factura;
        sig:listaFacturas;
    end;

{procedure cargarLista (var l:listaFacturas);} // se dispone

procedure eliminarSucursal (var l:listaFacturas;var cant:integer;var buscar:integer);
var
    act,ant:listaFacturas;
begin
    act := l;
    while (act<>nil) do
    begin
        if (act^.dato.codigo <> buscar) then
         begin
            ant:=act;
            act:=act^.sig;
         end
        else
         begin
          if (act:=l) then            // si tiene que borrar el primero
           begin
            l:=act^.sig
            ant:=l;
           end;
          else
           ant^.sig:=act^.sig;
        cant:=cant+1;
        dispose(act);
        act:=ant;
    end;
end;

var
 codigoEliminar,cantidad:integer;
 l:listaFacturas;

begin
  l:=nil;
  cargarLista(l); //se dispone
  cantidad:=0;
  writeln('Ingrese codigo de sucursal para eliminar sus facturas: ');
  readln(codigoEliminar);
  eliminarSucursal(l,cantidad,codigoEliminar);
  writeln ('se eliminaron ',cantidad,' facturas de la sucursal ',codigoEliminar);
end.
```

2. (No lo hice yo)



3)

Memoria estatica:

V = 121*4 = 484

Legajo = 8

I = 6

J = 6

Const dimF = 6

Total = 510 bytes

Memoria dinámica:

Estudiante = 60 bytes

For hace 101 new() = 101*60 = 6060

While hace 10 dispose () = 10 * 60 = 600

Total = 6060-600 = 5460

4) tiempo de ejecución:

i) 3*101+2+101*1 = 406 ut

ii) 1 ut

iii) 3*n+3 + 2*n donde n=10, es igual a 53 ut

Total = 406+1+53 = 460 ut

5)

a) Falso, si lo puede devolver porque es un tipo de dato simple

b) Falso, es valida, esta leyendo y asignándole un valor al contenido de la dirección de memoria a la que apunta el puntero. Aclaracion: es valida en tanto y en cuanto p apunte a un tipo de dato para el cual la operación read sea valida, por ejemplo si apunta a un registro es invalida.

c) Falso, pueden usarse ambas formas de comunicación entre modulos.

d) Falso, no necesariamente. No sabemos que son esas lineas de código, podría por ejemplo ser un for de 10000 iteraciones, y las 100 lineas de código 100 lecturas y escrituras. El tiempo de ejecución se calcula a partir de las operaciones elementales.

e) Falso, aunque todos sus datos sean del mismo tipo un registro es una estrucura de datos heterogenea.

f) Falso. Aunque la lista requiere un recorrido secuencial para llegar a sus elementos, si seria posible si por ejemplo guardásemos un puntero que apunte a ese elemento N o recorrremos la lista hasta llegar el elemento N (siempre y cuando este exista).