FINAL – OCTUBRE 2023

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

```
program FinalOctubre2023;
type
  codSucursal = 1..5;

  factura = record
    num: integer;
    codCli: integer;
    codSuc: codSucursal;
    monto: real;
  end;

  lista = ^nodo;
  nodo = record
    dato: factura;
    sig: lista;
  end;

procedure leerFactura(var f: factura) // SE DISPONE
procedure cargarLista(var pri: lista); // SE DISPONE
```

```
procedure eliminarFacturas(var cant, num: integer, var pri: lista);
var
  ant, act: lista;
begin
  act:= pri;
  while (act <> nil) do begin
    if (act^.dato.codSuc = num) then begin
      if (act = pri) then
        pri:= pri^.sig;
      else
        ant^.sig:= act^.sig;
      ant:= act^.sig;
      dispose(act);
      cant:= cant+1;
      act:= ant^.sig;
    end;
    ant:= act;
    act:= act^.sig;
  end;
end;

var
  cant, num: integer;
  pri: lista;
begin
  cant:= 0;
  write("Por favor ingrese un numero de sucursal: ")
  read(num);
  eliminarFacturas(cant, num, pri);
  writeln("Se eliminaron " + cant + " facturas de la sucursal " + num);
end.
```

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;
```

PROCESO 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[1]);
    v.dimL:= v.dimL - 1;
  end;
end;
```

Qué pasa con el for si la dimL = 0?

PROCESO B

```
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[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 el resultado y justificar.

```
program FinalOctubre2023;
const
  dimF = 120;
type
  cadena = string[45];// --> 46 bytes
  rango = 0..100; // --> 6 bytes

  estudiante = record
    ape_nom: cadena; // --> 46 bytes
    promedio: real; // --> 8 bytes
    leg: integer; // --> 6 bytes
  end; // --> total = 60 bytes

  vector = array [0..dimF] of ^estudiante;
  // --> 100 * 4 (puntero a estudiante) = 400 bytes

var
 v: vector, // --> 400 byes
 legajo: real; // --> 8 bytes
 i, j: integer; // --> 6+6 = 12 bytes
 // TOTAL MEMORIA ESTÁTICA = 420 bytes
begin
 for i:= 0 to 100 do begin
    new(v[i]); // primera posición errónea --> 100 * 60 = 600 bytes
    read(legajo);
    v[i]^.leg:= leg;
  end;
  j:= 20;
  while (j > 0) and (v[j].leg <> 1234) do begin
    dispose(v[j]); // peor caso primera iteración leg=1234?
    // TOTAL DINÁMICA --> 540 bytes ya que no se ejecutaría dispose
    j:= j-2;
  end;
end.
```

El programa no compila debido a que se está intentando asignar un real a una variable declarada como integer. En este caso el programa no se ejecuta y por lo tanto la memoria dinámica es 0?

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

```
program FinalOctubre2023;
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: integer;
begin
 for i:= 0 to 100 do begin // --> (3N + 2) + N(CUERPO) --> (3.100+2) + 100(1UT) = 302+100 = 400UT.
    new(v[i]);
    read(legajo);
    // cuenta la asignación para el cálculo de tiempo de ejecución si es errónea? asigna un real a un integer
    v[i]^.leg:= legajo;
  end;
  j:= 20; // --> 1UT
  while (j > 0) and (v[j]^.leg <> 1234) do begin // --> C(N+1) + N(CUERPO) --> 3.(10+1) + 10.2 = 33 + 20 = 53UT
    dispose(v[j]);
    j:= j-2;
  end;
  // TOTAL TIEMPO DE EJECUCIÓN = 454UT
end.
```

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.

      Falso. Un módulo función puede retornar cualquier tipo de datos simple, siendo el puntero uno de los mismos.

   b. Si p es una variable de tipo puntero, entonces no es válida la siguiente instrucción: readln(p^)

      Falso. Las operaciones read/write serán válidas dependiendo a qué tipo de dato esté apuntando el puntero. Por ejemplo, en el caso de que sea un puntero a integer podría usarse readln(p^) para leer su contenido, pero si fuera un puntero a registro habría que leer cada uno de sus campos por separado.

   c. Si un programa utiliza variables globales entonces no puede contener módulos que utilicen parámetros.

Falso. El uso de variables globales y el uso de parámetros son dos métodos de comunicación que pueden coexistir en un programa, sin embargo el uso de variables globales está desaconsejado.

d. Un módulo de 5 líneas de código es más eficiente en tiempo de ejecución que un programa con 100 líneas de código.

   Falso. El tiempo de ejecución siempre va a depender de lo que realice el programa. Tal vez el programa con 5 líneas de código hace que se imprima en pantalla un texto 30mil veces, lo cual podría demorar más que la tarea que realiza el 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.

   Falso. Que un registro pueda tener todos sus campos del mismo tipo es una coincidencia que no modifica la característica de la estructura de ser heterogénea.

f. Nunca es posible acceder al nodo de la posición N en una estructura del tipo lista.

   Falso. Si el nodo N existe en la lista y se guarda la posición por la cual se está iterando, entonces es posible acceder al nodo N.