

1. Un comercio dispone de una estructura de datos con las facturas (como máximo 2000 facturas) realizadas durante marzo de 2023. De cada factura se conoce el número de factura, código de cliente, código de sucursal y monto total. Las facturas se encuentran ordenadas por código de sucursal. Se pide implementar un programa con un módulo que reciba la estructura que se dispone y devuelva el código de sucursal con mayor cantidad de facturas. El programa debe informar el valor retornado por el módulo.
2. Dada la siguiente declaración y los siguientes procesos, indique para cada uno de los procesos si son correctos o no. El objetivo es duplicar el contenido del último nodo de la lista. Justifique su respuesta.

```

type miLista = ^nodo;
nodo = record
    dato: integer; sig: miLista;
end;
lista = record
    pri: miLista; ult: miLista;
end;

```

A	B
<pre> Procedure duplicar1 (L: lista); begin L.ult^.dato := L.ult^.dato * 2; end; </pre>	<pre> Procedure duplicar2 (L: lista); var aux: miLista; begin aux := L.pri; while (aux^.sig <> nil) do aux := aux^.sig; end; aux^.dato := aux^.dato * 2; end; </pre>

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.

<pre> program ejercicio3; const dimF = 200; type cadena31 = string[31]; alumno = record ape_nom: cadena31; promedio: real; end; vector = array [1..dimF] of ^alumno; lista = ^nodo; nodo = record datos: alumno; sig: lista; end; var v: vector; a: alumno; nota, i, suma, cant: integer; aux: lista; begin aux := nil; for i := 1 to dimF do begin read(a.ape_nom); read(nota); cant := 0; suma := 0; while (nota <> -1) do begin suma := suma + nota; cant := cant + 1; read(nota); end; if (cant <> 0) then a.promedio := suma/cant else a.promedio := 0; new (v[i]); v[i] := a; end; end; end. </pre>	<table> <tr> <td>Char</td><td>1 byte</td></tr> <tr> <td>Integer</td><td>4 bytes</td></tr> <tr> <td>Real</td><td>8 bytes</td></tr> <tr> <td>Boolean</td><td>1 byte</td></tr> <tr> <td>String</td><td>Longitud + 1 byte</td></tr> <tr> <td>Puntero</td><td>4 bytes</td></tr> </table>	Char	1 byte	Integer	4 bytes	Real	8 bytes	Boolean	1 byte	String	Longitud + 1 byte	Puntero	4 bytes
Char	1 byte												
Integer	4 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:

- Antes de utilizar una variable puntero siempre se debe reservar memoria.
- La comunicación mediante parámetros asegura que un programa es correcto.
- La invocación al módulo *otro* es válida.

program ejercicio;

var a, b: real;

procedure calcular (var x: real; c: real);

function otro (num: integer): integer;

begin

...

end;

begin

...

end;

begin

...

write (otro (40));

...

end.

d. Siempre es posible eliminar el primer elemento en una lista.

e. Las instrucciones dentro de una estructura de control **repeat... until()** se pueden ejecutar 0, 1 o más veces.

f. El siguiente programa muestra por pantalla:

Valor de a: 200 Valor de b: 30 Valor de c: 20

program imprimir;

var a, c: integer;

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

begin

x:=10; c:=c+b; a:=(b+x)*5; b:=(a+b) MOD 10;

end;

var b: integer;

begin

b:=20; c:=b-5;

calcular (b,c);

writeln('Valor de a: ', a, ' Valor de b: ', b, ' Valor de c: ', c);

end.

$$B = 20 / 2 = 10$$