

1. La Facultad de Informática organizara el congreso WICC, en donde se expondrán trabajos de investigación. Realizar un programa que lea la información de cada publicación: título de la publicación, nombre del autor, DNI del autor y tipo de publicación (1..12). La lectura de publicaciones finaliza al ingresar un DNI de autor con valor 0 (el cual no debe procesarse). La información se lee ordenada por DNI del autor y un autor puede tener varias publicaciones. Se pide escribir un programa que:
 - a. Informe el tipo de publicación con mayor cantidad de publicaciones.
 - b. Informar para cada autor la cantidad de publicaciones presentadas.
2. Realice el cálculo de la memoria estática del ejercicio resuelto en el ejer.1)
3. Indique y justifique qué se imprime en cada sentencia write:

```

program prueba;
Var c: integer;

Procedure Calcular (var b, a: integer);
Begin
  b:= 10;
  while (b>0) do begin
    a:= a+b;
    b:= b-2;
    c:= c-b;
  end;
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
End;
var
  a, b: integer;
begin
  b:= 80;
  c:= 30;
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
  calcular (a,b);
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
end.

```

4. Defina el tipo de dato Arreglo y describa los pasos necesarios (pseudocódigo) para realizar las operaciones de:
 - a. Buscar un elemento en un arreglo ordenado
 - b. Insertar un elemento en una posición determinada del arreglo
5. Calcule el tiempo de ejecución para el *procedure Calcular* del ejercicio 3).
6. Explique y ejemplifique las estructuras de control de selección vistas en el curso.
7. En caso de ser posible reescribir el código del *procedure Calcular* del ejer. 3) reemplazando la estructura iterativa *While* por la repetitiva *For*.

1. Un comercio dispone de una estructura de datos con las ventas (como máximo 500 ventas) realizadas durante el mes de enero. De cada venta se conoce el número de venta, monto total y una lista de los productos vendidos (de cada producto vendido se tiene código y cantidad). Se pide implementar un programa que elimine de la estructura de datos todas las ventas cuya lista de productos contengan exactamente 5 productos.
2. Mencione (solo nombre) todas las estructuras de control vistas en el curso y luego elija una de ellas. Para la estructura de control elegida, descríbala y escriba un enunciado que incluya un módulo que justifique su uso. Se debe implementar el módulo.
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;
type
  cadena43 = string[43];
  info = record
    nombre: cadena43;
    nota: integer;
  end;
  lista = ^ nodo;
  nodo = record
    datos: info;
    sig: lista;
  end;
  alumno = record
    ape_nom: cadena43;
    promedio: real;
  end;
  vector = array [1..10] of ^alumno;

```

```

var
  v: vector; a: alumno; i, dimL: integer; e: info; aux: lista;

```

```

begin
  1) dimL := 5;
  read(a.ape_nom);
  2) i := 0;
  3) while (i < 5) and (a.ape_nom <> 'ZZZ') do
    begin
      read(a.promedio);
      new (v[i]);
      4) v[i] := a;
      5) i := i + 1;
      read(a.ape_nom);
    end;
  end;
end.

```

Char	1 byte
Integer	6 bytes
Real	10 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. Las listas se clasifican como estructuras heterogéneas.
- b. Todos los módulos de tipo función de un programa se pueden escribir como módulos de tipo procedimiento.
- c. El siguiente programa muestra por pantalla: - Valor de a: 40 - Valor de b: 140

```
program ejercicio;
var
  a, b: integer; {v. global}

  procedure calculo(c: integer);
  begin
    b := a + 100;
    a := c + 10 + b;
  end;

begin
  a := 40;
  calculo(a+10);
  writeln('- Valor de a: ', a);
  writeln('- Valor de b: ', b);
end;
```

- d. Se puede agregar un elemento al final de una lista sin necesidad de realizar un recorrido secuencial de la misma.
- e. El tipo base de un tipo de dato subrango puede ser un tipo de dato real.
- f. Dado el encabezado siguiente: *Procedure Prueba (a: integer; var b: real);*
La siguiente invocación es correcta: *Prueba (8, 10);*
- g. La sección TYPE en un programa únicamente se puede incluir luego de la declaración de las constantes y antes de la declaración de los módulos.
- h. Un programa correcto con 3 módulos es más eficiente que un programa correcto con 5 módulos.

Apellido y Nombre: DNI:

1. **Práctica:** Se dispone de la información de los productos de un supermercado. De cada producto se tiene Código, Nombre, Rubro (1..20) y precio. Se pide implementar un programa que guarde en una estructura adecuada los productos de los rubros que tengan 10 productos.
2. Indique para las siguientes proposiciones, si son **Verdaderas o Falsas**. Justifique cada caso.
 - En la técnica de corrección de debugging es necesario analizar los casos límites del problema..
 - Un vector siempre se utiliza teniendo en cuenta la dimensión lógica.
 - Una función puede devolver un tipo de dato registro, real, booleano, integer, entre otros.
 - Un programa que utiliza sólo variables globales no requiere modularización.

3. Dada la siguiente declaración de tipos de datos y variables, justificar para cada sentencia numeradas son válidas o inválidas:

<pre> program ejercicio3; type cadena100 = string[100]; cliente = record codigo = integer; tel: integer; dir: cadena100; end; clientes = ^nodo; nodo = record datos: cliente; sig: clientes; end; var c: cliente; cli: clientes; </pre>	<pre> begin 1. read(c); 2. new(c); 3. read (cli); 4. c := nil; 5. cli := nil; 6. dispose(cli); 7. read(cli^.codigo); 8. write(c.codigo); end. </pre>
---	--

4. Describa las características de una estructura del tipo de dato vector y describa los pasos necesarios de la operación de búsqueda de un elemento en dicha estructura.
5. Teniendo en cuenta las referencias, calcule e indique la cantidad de **memoria estática** y el **tiempo de ejecución**. Muestre cómo obtiene resultado.

program ejercicio5;	Referencia	
<pre> type cadena20 = string[20]; notas = 2..10; alumno= record ape_nom: cadena20; nota: integer; end; vector = array [1..10] of ^alumno; var v: vector; i:integer; sum: integer; nota: notas; apeNom: cadena20; begin for i:= 1 to 10 do begin new(v[i]); read(nota); read (apeNom); v[i]^..nota:= nota; v[i]^..ape_nom:= apeNom; end; sum := 0; while (sum < 200) do begin read (nota); sum := sum + nota; end; end. </pre>	Char Integer Real Boolean String Puntero	1 byte 4 bytes 8 bytes 1 byte Longitud + 1 byte 4 bytes

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; 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.</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.

- Siempre es posible eliminar el primer elemento en una lista.
- Las instrucciones dentro de una estructura de control **repeat... until()** se pueden ejecutar 0, 1 o más veces.
- 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.

$$B = 20 / 2 = 10$$

2(008)

07/02/2023

Final Conceptos de Algoritmos Datos y Programas - TEMA 1

Apellido y Nombre [redacted]

1. Un comercio dispone de la información de sus clientes. De cada cliente se conoce su DNI, fecha y monto de la compra. La información está ordenada por el DNI del cliente. Se pide implementar un programa que genere una estructura de datos que contenga DNI del cliente, la cantidad de compras que realizó y el monto total de las compras, sólo con los clientes que realizaron más de 5 compras.

2. Dada la siguiente declaración y los siguientes módulos, indique para cada opción si incrementa en 1 cada uno de los elementos del vector. Justifique para cada caso.

```
Const dimF=50;
Type rango = 1..dimF;
tVector = array[rango] of ^integer;
vector = record
    vec: tVector;
    diml: integer;
end;
```

FACTA OPCIÓN A
La opción A
Incrementa el elemento del vector
a partir de la posición pasada por
el parámetro rango
vector

A	B
<pre>Procedure inicializar (v: vector); var i: rango; begin for i:= 1 to v.diml do begin new(v.vec[i]); v.vec[i]^:= v.vec[i]^ + 1; end; end;</pre>	<pre>Procedure inicializar (var v: vector); var i: rango; begin for i:= 1 to diml do begin v.vec[i]:= v.vec[i] + 1; end; 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.

```
program ejercicio3;
type
    cadena25 = string[25];
    lista = ^ nodo;
    nodo = record
        nombre: cadena25; nota: integer; sig: lista;
    end;
    alumno = record
        ape_nom: cadena25; materias: lista;
    end;
    vector = array [1..40] of ^alumno;
var
    v: vector; a: alumno; i: integer;
begin
    for i:= 20 downto 10 do begin
        read(a.ape_nom); a.materias:= nil;
        new (v[i]);
        v[i]^:= a;
    end;
    v[15]:= nil;
    dispose(v[10]);
end.
```

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

$3(n) + 2 + n(\text{cuerpo})$
 $3(10) + 2 + 10(2) \rightarrow 11$
 $30 + 2 + 20 = 52 \text{ UT}$
 Tiempo de ej. $\rightarrow 53 \text{ UT}$

1	B-
2	M
3	R
4	R+
5	R

Apellido y Nombre:

1. Indique qué imprime el siguiente código en Pascal.

```

program imprimir;
var
  a,b,c: integer;
procedure calcular (b: integer; var a: integer;)
begin
  c:= c + b; a:= (b + c) * 5; b:= (a + b) MOD 10;
  writeln(a,b,c);
end;
begin
  a:= 15; b:= 20; c:= b - a;
  calcular (a,c);
  writeln(a,b,c);
end.

```

2. Complete la columna de la izquierda con V o F según corresponda y justifique en
- todos**
- los casos:

<input type="checkbox"/>	a. Las instrucciones dentro de una estructura de control for siempre se ejecutan al menos 1 vez.
<input type="checkbox"/>	b. Un programa modularizado es eficiente.
<input type="checkbox"/>	c. En el acceso a los campos de un registro es necesario respetar el orden en que fueron declarados.
<input type="checkbox"/>	d. Una variable global sólo puede ser accedida y modificada desde el cuerpo del programa principal.
<input checked="" type="checkbox"/>	e. Para utilizar una variable de tipo puntero p siempre se debe realizar new (p) .
<input checked="" type="checkbox"/>	f. Se pueden utilizar operaciones de entrada/salida sobre todos los campos de una variable de tipo registro.
<input type="checkbox"/>	g. La estructura de datos Lista es heterogénea.
<input type="checkbox"/>	h. Al asignar el valor de nil a un puntero se libera la memoria referenciada.

3. Describa el tipo de dato vector y los pasos necesarios (pseudocódigo) para realizar la operación de insertar un vector en otro vector desde una posición determinada.

4. Realice el cálculo de la memoria estática y dinámica del siguiente programa.

Referencias: Integer (2), real (6), char (1), boolean (1) y puntero (4).

```

program ejercicio4;
const dimF = 10;
type
  cadenall= string[11];

  emple = record
    ape_nom: cadenall;
    edad: integer;
    sueldo: real;
  end;
vector = array[1..dimF] of ^emple;

```

```

var
  v: vector; e: emple; i: integer;
begin
  i:= 0;
  repeat
    i:= i + 1; new(v[i]);
    read(e.ape_nom, e.edad, e.sueldo);
    v[i]^:= e;
  until e.edad = 20;
  while (i > 0) do begin
    v[i]^sueludo:=v[i]^sueludo + v[i]^sueludo*0.25;
    i:= i - 1;
  end;
end.

```

5. Calcule el tiempo de ejecución para el programa del ejercicio 4).

Apellido y Nombre

1. Un comercio dispone de la información de sus clientes. De cada cliente se conoce su DNI, fecha y monto de la compra. La información está ordenada por el DNI del cliente. Se pide implementar un programa que genere una estructura de datos que contenga DNI del cliente, la cantidad de compras que realizó y el monto total de las compras, sólo con los clientes que realizaron más de 5 compras.

2. Dada la siguiente declaración y los siguientes módulos, indique para cada opción si incrementa en 1 cada uno de los elementos del vector. Justifique para cada caso.

```
Const dimF=50;
Type rango = 1..dimF;
tVector = array[rango] of ^integer;
vector = record
    vec: tVector;
    dimI: integer;
end;
```

A	B
<pre>Procedure inicializar (v: vector); var i: rango; begin for i:= 1 to v.dimI do begin new(v.vec[i]); v.vec[i]^:= v.vec[i]^ + 1; end; end;</pre>	<pre>Procedure inicializar (var v: vector); var i: rango; begin for i:= 1 to dimI do begin v.vec[i]:= v.vec[i] + 1; end; 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; type cadena25 = string[25]; lista = ^ nodo; nodo = record nombre: cadena25; nota: integer; sig: lista; end; alumno = record ape_nom: cadena25; materias: lista; end; vector = array [1..40] of ^alumno; var v: vector; a: alumno; i: integer; begin for i:= 20 downto 10 do begin read(a.ape_nom); a.materias:= nil; new (v[i]); v[i]^:= a; end; v[15]:= nil; dispose(v[10]); 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												

1	B-
2	M
3	R
4	R+
5	R

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. Antes de utilizar una variable puntero siempre se debe reservar memoria.
- b. La comunicación mediante parámetros asegura que un programa es correcto.
- c. 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.
```


Apellido y Nombre: DNI:

- Práctica:** Se dispone de la información de los participantes inscriptos a una carrera (a lo sumo 5000). De cada participante se tiene DNI, nombre y apellido, categoría (1..5) y fecha de inscripción. Se pide implementar un programa que guarde en una estructura adecuada los participantes de aquellas categorías que posean a lo sumo 50 inscriptos. Se sabe que cada participante se puede anotar en una sola categoría.
- Indique para las siguientes proposiciones, si son **Verdaderas** o **Falsas**. **Justifique** cada caso.
 - No es posible la utilización de las variables globales para la comunicación entre los módulos de un programa.
 - Siempre es posible realizar la eliminación de un elemento en un vector.
 - Un programa modularizado puede no ser correcto.
 - El acceso a un elemento de una estructura de datos lineal sólo es posible a través de un recorrido secuencial.
- Dada la siguiente declaración de tipos de datos y variables, justificar para cada sentencia numeradas son válidas o inválidas:

<pre> program ejercicio_3; type cadena50 = string[50]; cliente = record DNI: cadena50; ape_nom: cadena50; end; clientes = ^nodo; nodo = record dato: cliente; sig: clientes; end; var c: cliente; cli: clientes; cli_esp: clientes; </pre>	<pre> begin 1. read(c); 2. new(c); 3. cli := nil; 4. new(cli); 5. cli_esp := cli; 6. dispose(cli); 7. read(cli_esp^.DNI); 8. write(cli_esp^.DNI); end. </pre>
--	---

- Describa las formas de comunicación entre módulos vistas en la materia.
- Teniendo en cuenta las referencias, calcule e indique la cantidad de **memoria estática**, **memoria dinámica** y el **tiempo de ejecución**. Muestre cómo se obtienen los resultados.

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

<pre> program ejercicio_5; type cadena30 = string[30]; categorias = 1..5; participante = record ape_nom: cadena30; categ: categorias; tiempo: real; end; vector = array [1..20] of ^participante; var p: vector; i: integer; c: categorias; ayn: cadena30; begin for i:= 1 to 10 do begin new(p[i]); read(c); read(ayn); p[i]^.categ:= c; p[i]^.ape_nom:= ayn; p[i]^.tiempo:=0; end; for i:= 10 downto 5 do dispose(p[i]); end. </pre>	
--	--

Apellido y Nombre: [REDACTED]

LEGAJE [REDACTED]

TEMA 1

1. Realizar un programa que procese la información de productos de un supermercado. De cada producto se conoce: código, sector (1: limpieza, 2: bebidas ... 10: jardín), descripción, stock actual y precio. La información se encuentra almacenada en una estructura de datos adecuada sin ningún orden. Se pide informar los dos sectores con mayor cantidad de productos en stock.
2. Indique y justifique qué se imprime en cada sentencia writeln:

```

Program prueba;
Var
  c: integer;

Procedure calcular (a: integer; var b: integer);
Begin
  b:= 10;
  a:= a * 2;
  c:= c - b;
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
End;

var
  a, b: integer;
begin
  b:= 20;
  c:= 30;
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
  calcular (b, a);
  writeln ('a= ', a, 'b= ', b, 'c= ', c);
end.

```

3. Describa el tipo de dato Lista y los pasos necesarios (pseudocódigo) para realizar la operación de eliminar un elemento de la lista.
4. Explique las estructuras de control iterativas y de repetición vistas en el curso. Ejemplifique con una situación de uso para cada una.
5. Explique los problemas que pueden ocasionar el uso de variables globales en la comunicación entre módulos. Ejemplifique un caso en que el uso de parámetros evita alguno de estos problemas.
6. Realice el cálculo de la memoria estática y dinámica del siguiente programa.

Referencias: Integer (2), real (4), char (1), boolean (1) y puntero (4).

```

program ejercicio4;
type
  cadena15= string[15];

```

```

emple = record
  ape_nom: cadena15;
  edad: integer;
  sueldo: real;
end;

```

```

vector = array[1..10] of ^emple;

```

```

var
  v: vector; e: emple; i: integer;
begin
  for i:= 1 to 4 do begin .1
    new(v[i]); .2
    read(e.ape_nom, e.edad, e.sueldo); .3
    v[i]^:= e; +
  end;
  for i:= 3 to 9 do $
    v[i]^ .sueldo:=v[i]^ .sueldo + (v[i]^ .sueldo*0.25); &
  end.

```

7. Calcule el tiempo de ejecución para el programa del ejercicio 6).