FINAL CADP – 09/08/2022

Apellido y Nombre: ..... **DNI:** .....

1. **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.

2. 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.

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

```
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;
```

```
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.
```

4. Describa las formas de comunicación entre módulos vistas en la materia.

5. 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.

```
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.
```

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