

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. Incluir módulos dentro de un programa implica que el programa es más eficiente que otro programa que realiza la misma tarea pero sin utilizar módulos.

b. El siguiente programa es válido.

```
program ejercicio;  
  function auxiliar(val:integer): integer;  
  begin  
    val:= val * val;  
    auxiliar:= val;  
  end;  
  procedure calculo(c: Integer; var b:integer);  
  begin  
    b:= b + c DIV 4;  
  end;  
var  
  a,b:integer;  
begin  
  a:= 16;  
  b:= 6;  
  calculo(auxiliar(a),b);  
end.
```

c. No siempre es posible declarar un tipo subrango donde su tipo base sea cualquiera de los tipos simples en la teoría.

d. Un programa que utiliza un repeat until puede reescribirse utilizando un while.

e. La comunicación entre el programa y los módulos no sólo se puede hacer utilizando parámetros.

Apellido y Nombre

- Una empresa dispone de una estructura de datos con las ventas de su comercio. De cada venta se conoce número de venta, cantidad de productos y tipo de pago (efectivo o tarjeta). Se pide implementar un programa que genere una segunda estructura con las ventas cuya cantidad de productos tenga más dígitos pares que impares. En la estructura generada deben quedar almacenadas las ventas de tipo de pago efectivo antes que las de tipo de pago con tarjeta.
- Dados los siguientes programas indique para cada uno si son válidos o no. Además, analice si considera que el funcionamiento en ambos programas es el mismo o no. **JUSTIFIQUE**

A	B
<pre> Program uno; Type datos = array [1..100] of integer; Var d: datos; Begin //Operaciones e invocaciones a módulos para cargar y recorrer el vector d End;</pre>	<pre> Program dos; Var d: array [1..100] of integer; Begin // Operaciones e invocaciones a módulos para cargar y recorrer el vector d End;</pre>

- 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 info = record nombre: string; nota: integer; datos: ^integer; end; vector = array [1..100] of info; var v: vector; i,j: integer; e: info; begin read(e.nombre); i:=0; while (i <= 100) and (e.nombre <> 'ZZZ') do begin read(e.nota); e.datos:= nil; i:= i + 1; v[i]:= e; read(e.nombre); end; for j:= 1 to i do begin new(v[j].datos); v[j].datos:= v[j].nota MOD 10; end; end.</pre>	<pre> Char 1 byte Integer 6 bytes Real 10 bytes Boolean 1 byte String Longitud + 1 byte Puntero 4 bytes</pre>
--	--