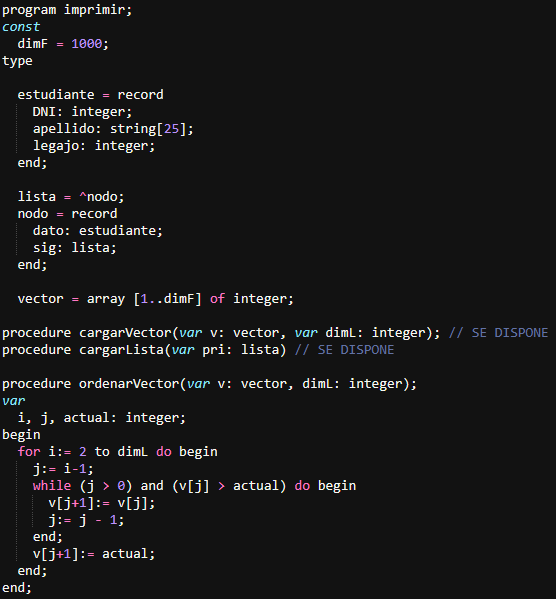
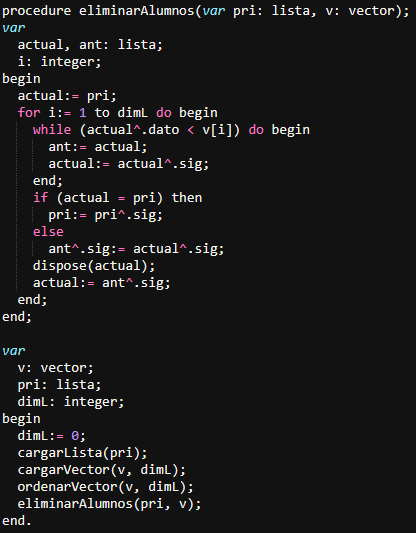
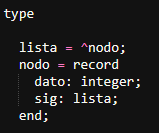
**FINAL – AGOSTO**

1. Una facultad dispone de los DNI de aquellos estudiantes que no cumplen con los requisitos de regularidad (a lo sumo 1000), los cuales no pueden seguir siendo estudiantes de la facultad. Además, dispone de una estructura en la cual almacena todos los estudiantes, de cada estudiante se conoce DNI, apellido y legajo. Esta estructura se encuentra ordenada por DNI. Se pide realizar un programa que elimine (eficientemente en tiempo de ejecución) de la estructura que posee todos los estudiantes aquellos estudiantes que han perdido regularidad.

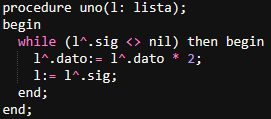




1. Teniendo en cuenta la siguiente declaración de tipos y los siguientes procesos, indique para cada uno de los procesos si los sueldos de los empleados se duplican de manera correcta (el valor del campo dato en cada nodo de la lista “l” recibida como parámetro). Justifique su respuesta.

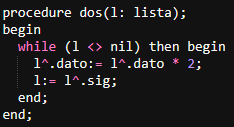


**PROCESO A**

****

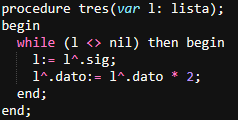
Este proceso es incorrecto. La condición del while no procesa el último nodo, la condición correcta para iterar hasta el final de la lista sería while (l <> nil) y la estructura de control está mal escrita, debido a que la manera correcta de escribir una estructura while es while (condición) do begin, no then begin.

**PROCESO B**

****

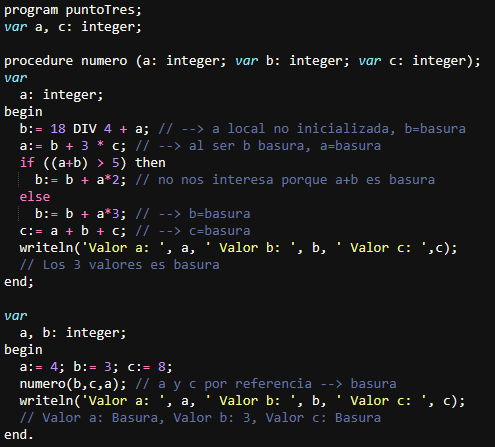
Este proceso es incorrecto. La estructura de control está mal escrita, ya que la forma correcta de escribir una estructura while es while (condición) do begin.

**PROCESO C**

****

Este proceso es incorrecto. En primera instancia la lista está pasada por parámetro por referencia, lo cual generaría que al recorrerla perdamos el valor del primer puntero a la misma. En segundo lugar, al avanzar al siguiente nodo antes de modificar el actual nos quedaría sin modificar el primer nodo de la lista. Por último, la estructura while está mal escrita, ya que su sintaxis correcta es while (condición) do begin.

1. Dado el siguiente programa indique que imprime en cada sentencia write. Justifique su respuesta.



1. Indique verdadero o falso. Justifique en todos los casos:
2. Si en un programa se encuentra la estructura de control IF siempre puede ser reemplazada por un CASE

Falso. El CASE solo puede evaluar variables de tipo ordinal, mientras que el IF permite evaluar también otros tipos de datos, como real o string, además de expresiones lógicas y aritméticas.

1. Agregar un elemento al final de una lista es menos eficiente en tiempo de ejecución que agregar un elemento en un arreglo.

Verdadero. El tiempo de ejecución que conlleva agregar atrás en una lista en el mejor caso es de 5UT, mientras que agregar en un vector conlleva 4UT.

1. La técnica de debugging puede aplicarse en cualquier instancia del desarrollo de un programa.

Verdadero. El debugging consiste en encontrar errores y corregirlos. Debido a que estos errores pueden ser sintácticos o semánticos, la corrección puede estar en diferentes etapas del desarrollo de un programa.

1. Si conozco la cantidad de elementos máxima que van a ser almacenados en una estructura siempre es más eficiente que esa estructura sea un vector.

Falso. Conocer la capacidad máxima de elementos no asegura realmente cuantos se van a almacenar, por lo tanto podríamos alocar más memoria estática de la necesaria si nos dejamos llevar por la cantidad máxima de elementos que puede haber. Esto resultaría en una pérdida de eficiencia respecto al consumo de memoria.

1. Para modificar los valores contenidos en una lista se puede utilizar un módulo que sea una función.

Verdadero. Las funciones solo aceptan parámetros por valor y como para modificar el contenido del nodo de una lista no es necesario pasar la variable como parámetro por referencia, una función podría modificar los valores contenidos en el nodo de una lista.