Nombre y Apellido: .....

Conceptos de Algoritmos Datos y Programas - 6/6/2017

1. **Práctica:** Realice un módulo que lee números enteros entre 0 y 100 y devuelva un vector que contenga la cantidad de ocurrencias de los valores leídos. La lectura finaliza cuando se lee el valor 0.

Ejemplo: si se leen los valores: 1, 20, 55, 7, 1, 7, 0 entonces el vector resultante deberá contener la información necesaria para saber que:
valor 1 cantidad de ocurrencias 2
valor 20 cantidad de ocurrencias 1
valor 55 cantidad de ocurrencias 1
valor 7 cantidad de ocurrencias 2

2. **Modularización**
a) Defina el concepto de Modularización y sus principales ventajas.
b) Explique las diferencias entre variable global, variable local, parámetro por valor y parámetro por referencia.
c) Dado el siguiente programa indique qué imprime en cada sentencia write, detallando los valores que toman las variables en cada paso:

```
program uno;
var
  a, b, c: integer;
procedure prueba (var a:integer; var b:integer; c:integer);
  var b: integer;
  Begin
    b:= a - 11;
    c:= b + 10;
    a:= a + c + 5;
    write (a, b, c);
  End;
```

```
Begin {programa principal}
  a:= 3;
  b:= 10;
  write (a, b, c);
  prueba (b, c, a);
  write (a, b, c);
End.
```

3. **Estructuras de Datos**
a) Defina y caracterice el tipo de dato Lista Enlazada. Especifique una representación posible en Pascal.
b) Describa detalladamente el problema de eliminar todas las ocurrencias de un valor en la estructura mencionada, teniendo en cuenta que la misma puede estar ordenada o no.

4. **Eficiencia**
a) Defina el concepto de eficiencia
b) Calcule la ocupación de memoria y el tiempo de ejecución para el siguiente código:

```
Program calculo;
Type
  cadena50 = string[50];
  persona= record
             nom : cadena50;
             edad: integer
           end;
lista = ^reg;
reg = record
        datos: persona;
        sig: lista;
      end;
Var
  Pri, aux: lista;
  p: persona;
  cant: integer;
```

```
begin
  pri:= Nil;
  read (p.nom, p.edad);
  while (p.nom <> 'ZZZ') do begin
    new (aux);
    aux^.datos := p;
    aux^.sig := pri;
    pri := aux;
    read (p.nom, p.edad);
  end;
  aux:= pri;
  cant:= 0;
  while (aux <> Nil) do begin
     if aux^.datos.edad = 18  then cant := cant + 1;
     aux:= aux^.sig
  end;
end.
```