

Explicación Parcial 02/07

TEMA 2
CADP 2022

Enunciado Tema 2

La cátedra de CADP está organizando el parcial, y necesita un programa para obtener información sobre los alumnos recursantes que rendirán.

- a) Realizar un módulo que **retorne** la información de los alumnos en una estructura de datos adecuada. Dicha información se lee por teclado sin ningún orden. De cada alumno se lee su DNI, nombre, apellido, año de ingreso, y nota obtenida (entre 0 y 10) **en cada una** de las 10 autoevaluaciones. En caso de no realizar alguna autoevaluación, se asigna la nota -1. La lectura finaliza cuando se ingresa el DNI 33016244 (que debe procesarse).
- b) Realizar un módulo que reciba la estructura de datos generada en el inciso anterior, e **IMPRIMA** en pantalla:
 - 1) DNI de los alumnos que podrán rendir el parcial de CADP. Para poder rendir, los alumnos deben haber realizado al menos 8 autoevaluaciones y deben haber aprobado (nota 6 o superior) al menos 4 autoevaluaciones.
 - 2) El porcentaje de alumnos que hayan ingresado en el año 2020 y que se hayan presentado a todas las autoevaluaciones, sobre el total de alumnos.
 - 3) Apellido y nombre de aquellos alumnos donde la suma de los dígitos del DNI es par.

NOTA: Implementar el programa principal.

Problema general

Se pide:

- A. Leer la información de cada alumno y almacenarla en una lista. El alumno es un **registro** con un campo de tipo vector (para las autoevaluaciones), entre otros. **No es necesario** tener una dimensión lógica para el vector de autoevaluaciones ya que siempre está cargado por completo.*

*La información se lee **desordenada** y no se requiere almacenar en ningún orden específico. Deberíamos hacer un insertar adelante o atrás que son los más simple y eficiente.*

*El corte de fin es 33016244 y **debe procesarse** por lo tanto hay que usar un repeat until para la lectura.*

Problema general

B. Los siguientes incisos se deben calcular e informar recorriendo una sólo vez la lista:

1. Imprimir DNI de los alumnos con cantidad de autoevaluaciones realizadas ≥ 8 y que aprobaron (nota ≥ 6) al menos 4. Es necesario **recorrer el vector** de autoevaluaciones. Si la nota es -1 significa que no realizó la autoevaluación.
2. Imprimir porcentaje de alumnos cuyo año de ingreso es 2020 y que se hayan presentado a todas las autoevaluaciones. Es necesario **recorrer el vector** de autoevaluaciones verificando que no haya ningún -1 (ausente). Hay que contar la cantidad de alumnos que cumplen y el total. Este punto se calcula al recorrer la lista y se imprime una única vez al final.
3. Imprimir nombre y apellido de los alumnos cuya suma de dígitos del dni es par. Descomponer cada DNI sumando sus dígitos en una variable. Verificar que el resultado sea par.

Solución Tema 2

```
program recursantes;
const
    cant_eval = 10;
type
    rango_eval = 1..cant_eval;
    vector_eval = array[rango_eval] of integer;

    alumno = record
        dni : integer;
        nombre_apellido : string;
        ingreso : integer;
        auto_eval : vector_eval;
    end;

    lista = ^nodo;
    nodo = record
        dato : alumno;
        sig : lista;
    end;
```

```
//PROGRAMA PRINCIPAL
var
    L: lista;
begin
    cargar_alumnos(L); //PUNTO A
    procesar_e_imprimir(L); //PUNTO B
end.
```

Solución Tema 2 - Inciso a

```
procedure leer_auto_eval(var v:vector_eval);  
var  
    i : rango_eval;  
begin  
    for i:= 1 to cant_eval do  
        readln(v[i]);  
    end;  
  
procedure leer_alumno(var a:alumno);  
begin  
    readln(a.dni);  
    readln(a.nombre_apellido);  
    readln(a.ingreso);  
    leer_auto_eval(a.auto_eval);  
end;
```

```
// PODRÍA AGREGAR ADELANTE, EN ESTE ENUNCIADO ES  
INDISTINTO  
procedure agregar_atras(var pri,ult : lista; a : alumno);  
var  
    aux : lista;  
begin  
    new(aux);  
    aux^.dato := a;  
    aux^.sig := nil;  
    if (pri=nil) then  
        pri := aux  
    else  
        ult^.sig := aux;  
    ult := aux;  
end;
```

Solución Tema 2 - Inciso a

```
procedure cargar_alumnos(var L : lista);  
var  
  a : alumno;  
  ult : lista;  
begin  
  l := nil;  
  repeat  
    leer_alumno(a);  
    agregar_atras(L,ult,a);  
  until (a.dni = 33016244)  
end;
```

Solución Tema 2 - Inciso b

```
procedure procesar_e_imprimir(L : lista);
var
    aprobadas, presentes, cant_alumnos, cant_total_alumnos : integer;
begin
    cant_alumnos := 0;
    cant_total_alumnos := 0;
    while (L <> nil) do
        begin
            cant_total_alumnos := cant_total_alumnos + 1;
            contar_auto_eval(L^.dato.auto_eval, presentes, aprobadas);
            //INCISO B1
            if (aprobadas >= 4) and (presentes >= 8) then
                writeln(L^.dato.dni);

            //INCISO B2
            if (cumple_condicion(L^.dato.ingreso, presentes)) then
                cant_alumnos := cant_alumnos + 1;

            //INCISO B3
            if (suma_digitos(L^.dato.dni) MOD 2 = 0) then
                writeln(L^.dato.nombre_apellido);
        end;
        writeln(porcentaje(cant_alumnos, cant_total_alumnos));
    end;
```


Solución Tema 2 - Inciso b

```
procedure contar_auto_eval(v:vector_eval; var presentes, aprobadas : integer);
var
    i : rango_eval;
begin
    presentes := 0;
    aprobadas := 0;
    for i:= 1 to cant_eval do
        if (v[i] >= 0) then
            begin
                presentes := presentes + 1;
                if (v[i] >= 6) then
                    aprobadas := aprobadas + 1;
                end;
            end;
        end;
    end;

function cumple_condicion(ingreso,presentes : integer) : boolean
begin
    cumple_condicion := (ingreso = 2020) and (presentes = cant_eval);
end;
```

Solución Tema 2 - Inciso b

```
function suma_digitos(num : integer) : integer;
var
    suma : integer;
begin
    suma := 0;
    while (num <> 0) do
    begin
        suma := suma + (num MOD 10); //sumo el último dígito de num
        num := num DIV 10;
    end;
end;

function porcentaje(num1, num2 : integer) : real
begin
    porcentaje := (num1 * 100 / num2);
end;
```