1. Función que reciba una matriz (N x N) de enteros y devuelva en una estructura de datos adecuada: el valor de la suma de los elementos de la diagonal principal y si es cierto o falso que los elementos de la diagonal son todos iguales.

Se pide:

- Definición de los tipos de datos adecuados *(1 punto)*
- Declaración *(1 punto)*y definición de la función*(3 puntos)*

Una solución:

```
Type
      tIndice = 1.. N;
      tMatrizEnteros = array [tIndice,tIndice] of integer;
      tRdiagonal = record
                        SonIguales: Boolean;
                        SumaDiagonal: integer
                       end;

Function valorDiagonal(protected Var  A: tMatrizEnteros): tRDiagonal;
Var r:tRDiagonal value [ SonIguales: true; SumaDiagonal: A[N, N]];
begin
      For i:= 1 to N -1 do begin
            If  A[i, i] <> A[i+1, i+1]
            Then r.SonIguales := false;
            r.SumaDiagonal:= A [i, i] + r.SumaDiagonal
      End;
      ValorDiagonal := r
End;
```

Apellidos, Nombre: *GrupoTeoría: 2 y 3*

2. Un número racional lo podemos representar como el cociente de dos enteros con denominador distinto de cero. Ejemplo: - ¾

Se pide:

- Definición de los tipos de datos adecuados para representar números racionales.*(1 punto)*
- Declaración.*(1 punto)* y definición de un procedimiento que sume dos números racionales y devuelva su resultado.*(3 puntos)*

Una solución:

```
Type
    tSigno = (mas, menos);
    tNatural = 0.. MaxInt;
    tpositivo = 1..MaxInt;

    tRacional = record
            signo : tSigno;
            numerador : tNatural;
            denominador : tPositivo;
            end;

function Signo(numero: tRacional): integer;
begin
  case numero.signo of
    mas   : Signo:= +1;
    menos : Signo:= -1;
  end
end;

procedure SumarFracciones(A,B: tRacional; var resultado: tRacional);
var num: integer;
begin

    resultado.denominador:= B.denominador * A.denominador;
    num:= Signo(A) * A.numerador * B.denominador + Signo(B) * B.numerador * A.denominador;
    resultado.numerador:= abs(num);

    if num < 0 then       resultado.signo:= menos
    else   resultado.signo:= mas

end;
```