


Ejemplo de código fuente de Unidades

12. Completar la unidad `race` con la siguiente sección de interfaz: 

```
type racional=record
    num,den:integer
end;
var cero:racional; (* cero debe ser la fraccion 0/1 *)
    uno:racional; (* uno debe ser la fraccion 1/1 *)
procedure asigna(var f:racional;n,d:integer);
function danum(f:racional):integer;
function daden(f:racional):integer;
procedure visualiza(f:racional);
procedure suma(f1,f2:racional;var f:racional);
```

El procedimiento `suma` debe incluir una llamada a un procedimiento privado (procedure `simplifica`) de la unidad que simplifique una fracción, convirtiéndola en irreducible.

```
(* La siguiente unidad declara el tipo de dato *)
(* racional e incorpora procedimientos para el *)
(* tratamiento numerico de fracciones *)
(* Autor: Division de Informatica Industrial *)
unit ejunit;
interface
    type racional=record
        num,den:integer
    end;
    var cero, uno: racional;
    procedure asigna(var f:racional;n,d:integer);
    function danum(f:racional):integer;
    function daden(f:racional):integer;
    procedure visualiza(f:racional);
    procedure suma(f1,f2:racional;var f:racional);
(* Seccion de implementacion *)
implementation
    procedure asigna;
    begin
        f.num:=n;
        f.den:=d
    end;
    function danum;
    begin
        danum:=f.num
    end;
    function daden;
    begin
        daden:=f.den
    end;
    procedure visualiza;
    begin
        write(f.num);
        write('/');
        writeln(f.den)
    end;
    procedure simplifica(var f:racional);
    var r,a,b:integer;
    begin
        a:=f.num;
        b:=f.den;
        repeat
            r:=a mod b;
            a:=b;
            b:=r
        until r=0;
        if a>1 then
            with f do begin
                num:=num div a;
```

```

        den:=den div a
    end
end;
procedure suma;
begin
    f.num:=f1.num*f2.den+f1.den*f2.num;
    f.den:=f1.den*f2.den;
    simplifica(f)
end;
(* Seccion de inicializacion *)
begin
with cero do
begin
num:=0;
den:=1
end;
with uno do
begin
num:=1;
den:=1
end
end.

```

13. Programa que hace uso de la anterior unidad `race`, sumando dos fracciones cualesquiera y visualizando su resultado por pantalla, incluyendo algún procedimiento que utilice una de las dos variables `cero` ó `uno`

```

(*****
(* Ejemplo de utilizacion de la unidad anterior *)
(* en un programa *)
(*****
(* Autor: Division de Informatica Industrial *)

program usoejuni;
uses race;
var g1,g2,g:racional;n,m:integer;
begin
writeln('intro numerador y denominador de la fraccion');
readln(n,m);
asigna(g1,n,m);
writeln('intro numerador y denominador de la fraccion');
readln(n,m);
asigna(g2,n,m);
suma(g1,g2,g);
visualiza(g);
suma(g,uno,g);
visualiza(g);
{ si se incluye la sentencia simplifica(g);
  se produce un error de compilacion por
  identificador desconocido
  ya que es una funcion privada }
end.

```