

**Fundamentos de la programación y la informática**  
**Examen de ejercicios de laboratorios. 11 de enero de 2024**  
URJC. Grados en ingeniería aeroespacial. Turno de tarde

---

- Recuerda que si el programa no compila, su nota será nula.

## Ejercicio único (10 puntos)

En el fichero

<https://gsyc.urjc.es/~mortuno/fpi/puzzle.pas>

encontrarás el esqueleto de un programa (`puzzle.pas`) cuyo objetivo es hacer lo siguiente:

- La función `genera_numero` recibirá como parámetro el entero `caras_dado`. Lanzará un dado de ese tamaño. La mitad de las veces devolverá el valor obtenido. La otra mitad, el valor multiplicado por -1. Para conseguir esto, tirará una *moneda*.
- El procedimiento `inicia_matriz` debe rellenar una matriz con los valores obtenidos de la función anterior. Excepto la última columna, que siempre la iniciará a 0. Ejemplo:

```
-2  2  2  0
 1  2 -5  0
-6  5 -6  0
```

- El procedimiento `rellena_puzzle` recorrerá una matriz, por filas. El último valor de cada fila (que se supone que será un 0, pero no hace falta que lo compruebes) lo reemplazará por un valor tal, que la suma de esa fila sea 0.

Ejemplo:

```
-2  2  2 -2
 1  2 -5  2
-6  5 -6  7
```

Observa que para conseguir esto, solo hay que ir sumando cada fila hasta el penúltimo elemento, y en el último, rellenar la suma obtenida, pero multiplicada por -1.

- El procedimiento `media_pos_neg` recibirá una matriz y
  - Calculará la media de los valores que sean positivos (considerando el 0 positivo).
  - Calculará la media de los valores que sean negativos.
  - Devolverá ambos valores en parámetros de salida.
- El procedimiento `informacion_matriz` recibirá como parámetro una matriz, y escribirá en pantalla:
  - La matriz.
  - La suma por filas de la matriz.
  - La media de los valores positivos.
  - La media de los valores negativos.

Observa que el cuerpo de este procedimiento ya está completo. No toques el cuerpo. Rellena lo que falta.

Observaciones:

1. No deberías modificar las funciones ya escritas en el esqueleto.
2. Puedes escribir todos los subprogramas que necesites.
3. Tu programa tiene que funcionar con matrices de cualquier tamaño. Pero no cambies las dimensiones de la matriz del esqueleto (3x4).
4. En tu cuenta encontrarás un ejecutable con la solución a este ejercicio. Obviamente no tienes el código fuente disponible, solo el ejecutable. Pero te servirá como referencia. Tu programa tiene que hacer lo mismo, ante cualquier duda, obsérvalo. Se valorará que el formato sea ese mismo o muy parecido.

## Solución

Encontrarás esta solución en

[https://gsyc.urjc.es/~mortuno/fpi/solucion\\_puzzle.pas](https://gsyc.urjc.es/~mortuno/fpi/solucion_puzzle.pas)

Antes de intentar resolver el ejercicio, puedes compilar el fichero, ejecutarlo y analizar el resultado para ayudarte a entender qué debe hacer el programa.

```
{$mode objfpc}{$H-}{$R+}{$T+}{$Q+}{$V+}{$D+}{$X-}{$warnings on}
```

```
program puzzle;
const
  Filas = 3;
  Columnas = 4;
type
  TipoMatriz = array[1..Filas, 1..Columnas] of integer;

function tira_dado(caras_dado:integer):integer;
begin
  result := random(caras_dado) + 1;
end;

procedure escribe_matriz(matriz:TipoMatriz);
var i,j: integer;
begin
  for i := 1 to Filas do begin
    for j:= 1 to Columnas do
      write(matriz[i,j]:4);
      writeln;
    end;
  end;
end;

procedure suma_por_filas(var matriz:TipoMatriz);
var i,j, sumatorio: integer;
begin
  for i:= 1 to Filas do begin
    sumatorio := 0;
    for j:= 1 to Columnas do
      sumatorio := sumatorio + matriz[i,j];
    write(sumatorio:4);
  end;
  writeln();
end;

function genera_numero(caras_dado:integer):integer;
var
  rval:integer;
begin
  rval := tira_dado(caras_dado);
  if tira_dado(2) = 1
  then
    rval := rval * -1;
  result := rval;
end;

procedure inicia_matriz(var matriz:TipoMatriz);
var
  i,j : integer;
const
  CarasDado = 6;
begin
  for i := 1 to Filas do begin
    for j:= 1 to Columnas-1 do
      matriz[i,j] := genera_numero(CarasDado);
    matriz[1,Columnas] := 0;
  end;
end;
```

```

    end;
end;

procedure rellena_puzzle(var matriz:TipoMatriz);
var i,j, sumatorio: integer;
begin
    for i:= 1 to Filas do begin
        sumatorio := 0;
        for j:= 1 to Columnas -1 do
            sumatorio := sumatorio + matriz[i,j];
            matriz[i,columnas] := sumatorio * -1;
        end;
    end;
end;

function media(sumatorio, contador: integer):real;
begin
    if contador <> 0 then
        result := sumatorio / contador
    else
        result := 0
    end;
end;

procedure media_pos_neg(var matriz: TipoMatriz; var media_pos, media_neg: real);
var i,j: integer;
cont_pos, cont_neg, sum_pos, sum_neg: integer;
begin
    cont_pos := 0;
    cont_neg := 0;
    sum_pos := 0;
    sum_neg := 0;

    for i:= 1 to Filas do
        for j:= 1 to Columnas do begin
            if matriz[i,j] >= 0 then begin
                cont_pos := cont_pos + 1;
                sum_pos := sum_pos + matriz[i,j];
            end;
            if matriz[i,j] < 0 then begin
                cont_neg := cont_neg + 1;
                sum_neg := sum_neg + matriz[i,j];
            end;
        end;

        media_pos := media(sum_pos, cont_pos);
        media_neg := media(sum_neg, cont_neg);
    end;

procedure informacion_matriz(var matriz: TipoMatriz);
var
    media_pos, media_neg : real;
begin
    writeln('Matriz:');
    escribe_matriz(matriz);
    writeln();

    writeln('Suma por filas:');
    suma_por_filas(matriz);
    writeln();

    media_pos_neg(matriz, media_pos, media_neg);
    writeln('Media valores positivos:', media_pos:5:2);
    writeln('Media valores negativos:', media_neg:5:2);
end;

```

```
var
  matriz : TipoMatriz;
begin
  randomize();
  inicia_matriz(matriz);
  informacion_matriz(matriz);
  writeln();

  rellena_puzzle(matriz);
  informacion_matriz(matriz);
end.
```