

Programación de funciones que trabajan con arreglos y utilizan ciclos for en el lenguaje de MATLAB

Objetivos. Aprender a definir funciones en el lenguaje de MATLAB que trabajen con arreglos y utilicen ciclos.

Requisitos. Se supone que el estudiante ya tiene experiencia de definir funciones nuevas, trabajar con arreglos y escribir ciclos for en MATLAB o en alguno de sus análogos libres (GNU Octave, Scilab, FreeMat).

En los siguientes ejemplos se trata de GNU Octave, pero casi lo mismo se aplica en MATLAB, Scilab y FreeMat.

1. Ejemplo de una función que produce una lista de números usando un ciclo for. Definimos una sucesión $(x_k)_{k=1}^{\infty}$ mediante una recurrencia lineal de segundo orden y dos condiciones iniciales:

$$\begin{aligned}x_k &= 2x_{k-1} + x_{k-2} & (k \geq 3), \\x_1 &= 3, & x_2 = 5.\end{aligned}$$

La siguiente función construye un arreglo (columna) con los primeros n elementos de esta sucesión (se supone que $n \geq 2$). En cualquier editor crear el siguiente archivo de texto y guardarlo en el formato “texto simple” con el nombre `mysequence.m`:

```
function [x] = mysequence(n),
    x = zeros(n, 1);
    x(1) = 3;
    x(2) = 5;
    for k = 3 : n,
        x(k) = 2 * x(k - 1) + x(k - 2);
    endfor
endfunction
```

Abrir GNU Octave, usando los comandos `cd`, `pwd` y `ls` moverse a la carpeta donde está guardado el archivo `harm.m` y ejecutar los siguientes comandos:

```
mysequence(5)
mysequence(8)
```

2. Ejemplo de una función que produce una matriz usando un ciclo for.

```
function [a] = flipmatrix(n),
    a = zeros(n);
    for k = 1 : n,
        a(k, n + 1 - k) = 1;
    endfor
endfunction
```

Pruebas:

```
flipmatrix(3)
flipmatrix(5)
```

3. Ejemplo de una función que cuenta el número de entradas positivas en un arreglo dado con un ciclo for.

```
function [c] = countpositives(a),
    c = 0;
    for k = 1 : length(a),
        if a(k) > 0,
            c += 1;
        endif
    endfor
endfunction
```

Aquí en vez del operador condicional se puede usar la siguiente construcción porque GNU Octave identifica `true` con el número 1:

```
function c = countpositives(a),
    c = 0;
    for k = 1 : length(a),
        c += a(k) > 0;
    endfor
endfunction
```

Pruebas:

```
countpositives([7; -3; 5; 6; 4])
a = rand(1, 10);
countpositives(a)
```

4. Consejo para futuro: evitar el ciclo for. Espero que los ejemplos escritos arriba puedan ser útiles para los estudiantes que están aprendiendo la programación en el lenguaje de MATLAB. Por supuesto, en todos estos ejemplos hay soluciones más eficientes que utilizan operaciones matriciales y evitan el ciclo for.