

# Teoría de Autómatas y Lenguajes Formales

## Práctica 3

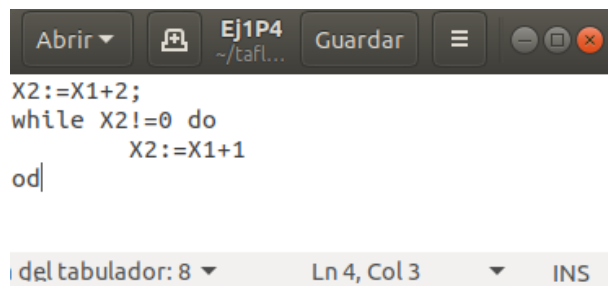
Marina González Torres

### Ejercicio 1

**Cree el programa WHILE más simple que calcule la función de divergencia (con cero argumentos) y calcule la codificación de su código.**

Para realizar este ejercicio vamos a usar el fichero proporcionado llamado "CODE2N". Dicho fichero nos va a devolver la codificación del código WHILE que le pasemos como parametro.

Para que el código diverja tenemos que realizar un bucle infinito.

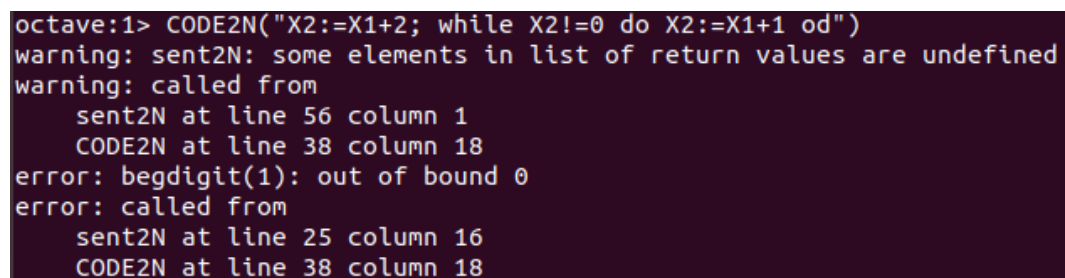


```
Abrir ▾ Ej1P4 ~/tafl... Guardar ≡ ⌵ □ ×
```

```
X2:=X1+2;  
while X2!=0 do  
    X2:=X1+1  
od|
```

```
del tabulador: 8 ▾ Ln 4, Col 3 ▾ INS
```

Sin embargo, al ejecutarlo con Octave se produce un error sin saber muy bien el motivo:



```
octave:1> CODE2N("X2:=X1+2; while X2!=0 do X2:=X1+1 od")  
warning: sent2N: some elements in list of return values are undefined  
warning: called from  
    sent2N at line 56 column 1  
    CODE2N at line 38 column 18  
error: begdigit(1): out of bound 0  
error: called from  
    sent2N at line 25 column 16  
    CODE2N at line 38 column 18
```

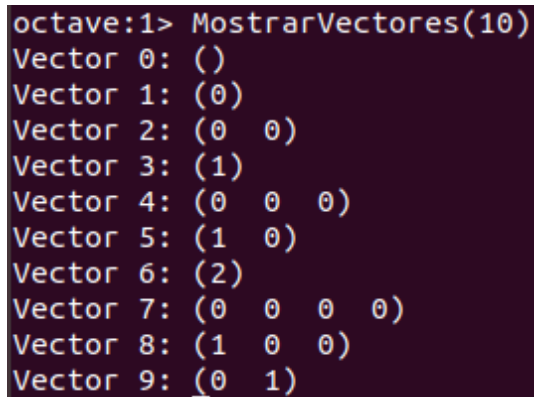
## Ejercicio 2

**Cree un script de Octave que enumere todos los vectores.**

Para crear el scrip también usaremos el script dado llamado "godeldecoding". El script será el siguiente:

```
function MostrarVectores(N)
for i=0: N-1
    fprintf('Vector %s: (%s)\n', num2str(i), num2str(godeldecoding(i)))
end
end
```

Un ejemplo de su ejecución con Octave sería el siguiente:



```
octave:1> MostrarVectores(10)
Vector 0: ()
Vector 1: (0)
Vector 2: (0 0)
Vector 3: (1)
Vector 4: (0 0 0)
Vector 5: (1 0)
Vector 6: (2)
Vector 7: (0 0 0 0)
Vector 8: (1 0 0)
Vector 9: (0 1)
```

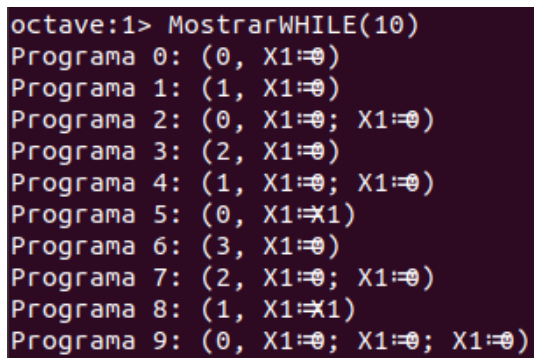
## Ejercicio 3

**Cree un script de Octave que enumere todos los programas WHILE.**

Para crear este script usaremos "N2WHILE" que es un script ya proporcionado. El script quedaría de la siguiente forma:

```
function MostrarWHILE(N)
for i=0: N-1
    fprintf('Programa %s: %s\n', num2str(i), N2WHILE(i))
end
end
```

Un ejemplo de su ejecución con Octave sería el siguiente:



```
octave:1> MostrarWHILE(10)
Programa 0: (0, X1:=0)
Programa 1: (1, X1:=0)
Programa 2: (0, X1:=0; X1:=0)
Programa 3: (2, X1:=0)
Programa 4: (1, X1:=0; X1:=0)
Programa 5: (0, X1:=X1)
Programa 6: (3, X1:=0)
Programa 7: (2, X1:=0; X1:=0)
Programa 8: (1, X1:=X1)
Programa 9: (0, X1:=0; X1:=0; X1:=0)
```

El igual del WHILE se ve desplazado aunque desconozco el motivo.