

1 Desarrollo del cálculo de la menor codificación del programa WHILE "diverger".

```
Q = (s, 0)
s :
X2 := X1 + 1
while X2 ≠ 0 do
  X1 := 0
od
```

La idea es asignar a una de las variables un número no nulo y entrar en un bucle que no termine hasta que dicha variable se haga nula, y en el código del bucle no modificarla. De esta manera el bucle diverge y la codificación es la menor.

```
>> CODE2N("X2=X1+1;while X2≠0 do X1=0 od")
ans = 10876
```

2 Código Octave que hace un print de todos los vectores, y una captura de ejemplo de ejecución.

```
{
function practica42()
    i=0
    while(i>=0)
        disp(['(' num2str(godeldecoding(i)) ')'])
        i+=1
    end
end
}
```

,

3 Código Octave que hace un print de todos los programas WHILE, y una captura de ejemplo de ejecución.

```
{
function practica43()
    i=0
    while( i>=0)
        disp(N2WHILE(i))
        i+=1
    end
end
}
```

,

```
>> practica42()
i = 0
()
i = 1
(0)
i = 2
(0 0)
i = 3
(1)
i = 4
(0 0 0)
i = 5
(1 0)
i = 6
(2)
i = 7
(0 0 0 0)
i = 8
(1 0 0)
i = 9
(0 1)
i = 10
(3)
i = 11
(0 0 0 0 0)
i = 12
(1 0 0 0)
i = 13
(0 0 1)
i = 14
(2 0)
i = 15
(4)
i = 16
(0 0 0 0 0 0)
i = 17
```

```
>> practica43()
i = 0
(0, X1=0)
i = 1
(1, X1=0)
i = 2
(0, X1=0; X1=0)
i = 3
(2, X1=0)
i = 4
(1, X1=0; X1=0)
i = 5
(0, X1=X1)
i = 6
(3, X1=0)
i = 7
(2, X1=0; X1=0)
i = 8
(1, X1=X1)
i = 9
(0, X1=0; X1=0; X1=0)
i = 10
(4, X1=0)
i = 11
(3, X1=0; X1=0)
i = 12
(2, X1=X1)
i = 13
(1, X1=0; X1=0; X1=0)
i = 14
(0, X1=X1; X1=0)
i = 15
(5, X1=0)
i = 16
(4, X1=0; X1=0)
i = 17
```