

Práctica 4

Juan José Ruiz Cañizal

26 de diciembre de 2022

1. Ejercicio 1

El programa WHILE más simple sin argumentos que calcula la función divergencia:

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

es:

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

De forma que al desarrollarlo por partes obtenemos que:

$$\Gamma(sentence2N(X2 := X1 + 1), sentence2N(while X2! = 0 do X1 := 0 od)) - 1 = 10876$$

Ya que:

$$sentence2N(X2 := X1 + 1) + sentence2N(while X2! = 0 do X1 := 0 od) = 10877$$

2. Ejercicio 2

```
function printvectors()
for i=0;
while(i>=0)
  disp(['(' num2str(godeldecoding(i)) ')'])
  i=i+1;
endwhile
end
```

Esta función itera sobre num2str, que devuelve un código dado un número, y godeldecoding degodeliza los vectores, por tanto solo hay que iterar sobre el vector de longitud N.

```

Archivo Editar Ver Buscar Terminal Ayuda
Read http://www.octave.org/bugs.html to learn how to submit bug reports.
For information about changes from previous versions, type 'news'.

octave:1> printVectors()
()
(0)
(0 0)
(1)
(0 0 0)
(1 0)
(2)
(0 0 0 0)
(1 0 0)
(0 1)
(3)
(0 0 0 0 0)
(1 0 0 0)
(0 0 1)
(2 0)
(4)
(0 0 0 0 0 0)
(1 0 0 0 0)
(0 0 0 1)
(0 1 0)
(1 1)
(5)
(0 0 0 0 0 0 0)

```

3. Ejercicio 3

```

function code = printwhile()
for i=0;
while (i>=0)
    disp(N2WHILE(i))
    i=i+1;
endwhile
end

```

Este caso es muy parecido al ejercicio anterior, solo que ne vez de hacer la biyección de N a strings, esta vez necesitaremos un codigo que haga una biyección de N a WHILE.

```

Archivo Editar Ver Buscar Terminal Ayuda
octave:1> printwhile()
(0, X1=0)
(1, X1=0)
(0, X1=0; X1=0)
(2, X1=0)
(1, X1=0; X1=0)
(0, X1=0; X1=0)
(3, X1=0)
(2, X1=0; X1=0)
(1, X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(4, X1=0)
(3, X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(5, X1=0)
(4, X1=0; X1=0)
(3, X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(6, X1=0)
(5, X1=0; X1=0)
(4, X1=0; X1=0)
(3, X1=0; X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(7, X1=0)
(6, X1=0; X1=0; X1=0)
(5, X1=0; X1=0; X1=0)
(4, X1=0; X1=0; X1=0)
(3, X1=0; X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(8, X1=0)
(7, X1=0; X1=0; X1=0)
(6, X1=0; X1=0; X1=0)
(5, X1=0; X1=0; X1=0)
(4, X1=0; X1=0; X1=0)
(3, X1=0; X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)
(9, X1=0)
(8, X1=0; X1=0; X1=0)
(7, X1=0; X1=0; X1=0)
(6, X1=0; X1=0; X1=0)
(5, X1=0; X1=0; X1=0)
(4, X1=0; X1=0; X1=0)
(3, X1=0; X1=0; X1=0)
(2, X1=0; X1=0; X1=0)
(1, X1=0; X1=0; X1=0)
(0, X1=0; X1=0; X1=0)

```

