

Práctica 4

Araceli Ruiz Vallecillo

26 de diciembre de 2022

Ejercicio 1

En este ejercicio vamos a crear el **programa WHILE** mas simple que diverja con cero argumentos de entrada:

$Q = (0, s)$

s:

```
 $X_2 := X_1 + 1;$   
while  $X_2 \neq 0$  do  
     $X_1 := 0$   
od
```

A la hora de ejecutar el codigo en octave podemos ver como el código diverge, este lo hemos probado en el script **F-emulation**:

```
octave:3> F_emulation("(1, X2:=X1+1; while X2!=0 do X1:=0 od)", 0)  
^C  
octave:4>
```

Hemos visto que no acaba y que hemos tenido que cortar la ejecución de este, por lo que diverge. La codificación de dicho código será la siguiente:

```
octave:10> CODE2N("X2:=X1+1;while X2!=0 do X1:=0 od")  
ans = 10876  
octave:11>
```

Ejercicio 2

En este ejercicio vamos a crear un **codigo en el que imprimamos N vectores de tamaño variado**.

El codigo implementado sería el siguiente:

```
function retval = printNvectors (N)

    for i = 0: N-1
        disp(['(' num2str(godeldecoding(i)) ')'])
    end

end
```

Por otro lado la imagen de dicho código será:

```
## Copyright (C) 2022 Ara
##
## This program is free software: you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation, either version 3 of the License, or
## (at your option) any later version.
##
## This program is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
## GNU General Public License for more details.
##
## You should have received a copy of the GNU General Public License
## along with this program. If not, see <https://www.gnu.org/licenses/>.
##
## *- texinfo -*
## @deftypefn {} (@var{retval} =) printNvectors (@var{input1}, @var{input2})
##
## @seealso{}
## @end deftypefn
##
## Author: Ara <usuario@usuario-VirtualBox>
## Created: 2022-12-24

function retval = printNvectors (N)

    for i = 0: N-1
        disp(['(' num2str(godeldecoding(i)) ')'])
    end

end
```

A continuación vamos a testear su comportamiento en Octave:

```
octave:1> printNvectors(5)
()
(0)
(0 0)
(1)
(0 0 0)
octave:2>
```

Como hemos podido comprobar nos salen 5 vectores, tal y como habíamos pedido por entrada.

Ejercicio 3

En este ejercicio vamos a crear una **función capaz de calcular N programas While**.

El código implementado sería el siguiente:

```
function retval = printNWhileprograms (N)

    for i = 0: N-1
        disp(N2WHILE(i))
    end

end
```

Por otro lado la imagen de dicho código será:

```
## Copyright (C) 2022 Ara
##
## This program is free software: you can redistribute it and/or modify
## it under the terms of the GNU General Public License as published by
## the Free Software Foundation, either version 3 of the License, or
## (at your option) any later version.
##
## This program is distributed in the hope that it will be useful,
## but WITHOUT ANY WARRANTY; without even the implied warranty of
## MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
## GNU General Public License for more details.
##
## You should have received a copy of the GNU General Public License
## along with this program. If not, see <https://www.gnu.org/licenses/>.
##
## -*- texinfo -*-
## @deftypefn {} (@var{retval} =) printNWhileprograms (@var{input1}, @var{input2})
##
## @seealso{}
## @end deftypefn
##
## Author: Ara <usuario@usuario-VirtualBox>
## Created: 2022-12-24
function retval = printNWhileprograms (N)
    for i=0:N-1
        disp(N2WHILE(i))
    end
end
end
```

Para ver la salida por pantalla de dicha función utilizamos **'Octave'**:

```
octave:2> printNWhileprograms (3)
(0, X1:=0)
(1, X1:=0)
(0, X1:=0; X1:=0)
octave:3> 
```

Como podemos ver la salida por pantalla da el número de programas While que hemos pedido por pantalla.