

# Práctica 4

Jesús Fuentes Moya

3 de diciembre de 2022

# Índice general

1. Desarrollo del cálculo de la menor codificación del programa WHILE “diverger”.	3
2. Código Octave que hace un print de todos los vectores, y una captura de ejemplo de ejecución.	4
3. Código Octave que hace un print de todos los programas WHILE, y una captura de ejemplo de ejecución.	6

# Capítulo 1

## Desarrollo del cálculo de la menor codificación del programa WHILE “diverger”.

Para obtener la menor codificación del programa while que diverge hemos pensado en buscar uno con el menor número de variables posibles y un índice bajo. Finalmente hemos determinado que el programa while “diverger” con menor codificación es:

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

cuya codificación del código es: 139126.

Hemos confirmado que este programa diverge usando “F.m”.

## Capítulo 2

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

Para realizar este script he usado el script del repositorio “godeldecoding.m” este script devuelve un vector asociado al numero introducido como argumento. Por tanto, para realizar el script he usado un bucle while cuyo contador se incrementa indefinidamente, y en cada iteración llamamos al script “godeldecoding.m” donde el parámetro utilizado es i. Por ende, el script obtenido es:

```
function ej = ejercicio1()
    i=0
    while(i!=-1)
        ej=godeldecoding(i)
        i=i+1
        printf("\n")
    end
end
```

cuya ejecución es:

```

>> ejercicio1
i = 0
ej = [](0x0)
i = 1

ej = 0
i = 2

ej =

    0    0

i = 3

ej = 1
i = 4

ej =

    0    0    0

i = 5

ej =

    1    0

i = 6

ej = 2
i = 7

ej =

    0    0    0    0

i = 8

ej =

    1    0    0

```

## Capítulo 3

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

Para elaborar este script vamos a usar el script “N2WHILE.m”. El script que se nos pide realizar es totalmente análogo al anterior. Lo único que en vez de usar el script “godeldecoding.m” usaremos el script “N2WHILE.m”. Obteniendo el siguiente script:

```
function ej = ejercicio2()
    i=0
    while(i!=-1)
        ej=N2WHILE(i)
        i=i+1
        printf("\n")
    end
end
```

cuya ejecución es:

```

>> ejercicio2
i = 0
ej = (0, X1=0)
i = 1

ej = (1, X1=0)
i = 2

ej = (0, X1=0; X1=0)
i = 3

ej = (2, X1=0)
i = 4

ej = (1, X1=0; X1=0)
i = 5

ej = (0, X1=X1)
i = 6

ej = (3, X1=0)
i = 7

ej = (2, X1=0; X1=0)
i = 8

ej = (1, X1=X1)
i = 9

ej = (0, X1=0; X1=0; X1=0)
i = 10

ej = (4, X1=0)
i = 11

```