

;Estos programas para la máquina de Turing son útiles para calcular
;la complejidad computacional de la suma de comprobación de IPv4
;propuestas por Comer y Stevens, así como lo propuesto por las
;RFC's que lo mencionan

; Suma de dos binarios en aritmética de complemento a 1. Suma intermedia COMER

; Entradas: dos números binarios, separados por un espacio, pe. '100 111'

; Programó: Roberto Cárdenas

; Fecha: febrero 22 de 2022

0 * * l qx

qx * * l qz

qz * * l qp

qp * s r qy

qy * * r qm

qm * * r qq

qq _ _ r 1

qq * * r q0

q0 _ _ r 1

q0 * * r q0

1 _ _ l 2

1 * * r 1

2 0 _ l 3x

2 1 _ l 3y

2 _ _ l 7

3x _ _ l 4x

3x * * l 3x

3y _ _ l 4y

3y * * l 3y

4x 0 x r q0

4x 1 y r q0

4x _ x r q0

4x * * l 4x ; skip the x/y's

4y 0 1 * 5

4y 1 0 l 4y

4y _ 1 * 5

4y * * l 4y ; skip the x/y's

5 x x l 6
5 y y l 6
5 _ _ l 6
5 * * r 5
6 0 x r q 0
6 1 y r q 0

7 x 0 l 7
7 y 1 l 7
7 _ _ l 10
7 * * l 7

;no hay carry
10 _ * l 99
99 s _ r 100
99 _ * r 150
150 _ * r 150
150 0 _ r 3000
100 _ * r 100
100 0 0 r 3000
100 1 1 r 3000

;detectar carry
10 * * * 12
12 s _ r 13
13 _ _ r 8

;SUMAR CARRY
8 1 o r 9
9 1 u r 9
9 0 o r 9

;Llegar a separador
9 _ * l 900
900 * * l 900
900 u * * 1001
900 o * * 1020
900 _ * r 2001

;Lee u

1001 u 1 r 1002

;Lee o

1020 o 0 r 1002

;escribe q y se devuelve

1002 * * r 1002

1002 _ * r 1003

1003 * * r 1003

1003 _ q l 1004

1004 * * l 1004

1004 o * * 1020

1004 u * * 1001

1004 _ * l 900

;última vuelta al comienzo

2001 * * r 2001

2001 q 0 r 2002

2002 q 0 r 2002

2002 _ * l 2003

2003 0 1 l 2004

2004 * * l 2004

2004 _ * l 2005

2005 * * l 2005

2005 _ * r q0

;Checksum COMER

3000 1 1 r 3000

3000 0 0 r 3000

3000 _ * * halt

; Cálculo del Checksum de acuerdo con COMER. Suma final
; Entradas: dos número binarios, separados por un espacio, pe. '100 111'
; Programó: Roberto Cárdenas
; Fecha: febrero 22 de 2022

0 * * l qx
qx * * l qz
qz * * l qp
qp * s r qy
qy * * r qm
qm * * r qq
qq _ _ r 1
qq * * r q0
q0 _ _ r 1
q0 * * r q0

1 _ _ l 2
1 * * r 1

2 0 _ l 3x
2 1 _ l 3y
2 _ _ l 7
3x _ _ l 4x
3x * * l 3x
3y _ _ l 4y
3y * * l 3y
4x 0 x r q0
4x 1 y r q0
4x _ x r q0
4x * * l 4x ; skip the x/y's
4y 0 1 * 5
4y 1 0 l 4y
4y _ 1 * 5
4y * * l 4y ; skip the x/y's
5 x x l 6
5 y y l 6
5 _ _ l 6
5 * * r 5

6 0 x r q0

6 1 y r q0

7 x 0 l 7

7 y 1 l 7

7 _ _ l 10

7 * * l 7

;no hay carry

10 _ * l 99

99 s _ r 100

99 _ * r 150

150 _ * r 150

150 0 _ r 3000

100 _ * r 100

100 0 1 r 3000

100 1 0 r 3000

;detectar carry

10 * * * 12

12 s _ r 13

13 _ _ r 8

;SUMAR CARRY

8 1 o r 9

9 1 u r 9

9 0 o r 9

;Llegar a separador

9 _ * l 900

900 * * l 900

900 u * * 1001

900 o * * 1020

900 _ * r 2001

;Lee u

1001 u 1 r 1002

;Lee o

1020 o 0 r 1002

;escribe q y se devuelve

1002 * * r 1002

1002 _ * r 1003

1003 * * r 1003

1003 _ q l 1004

1004 * * l 1004

1004 o * * 1020

1004 u * * 1001

1004 _ * l 900

;última vuelta al comienzo

2001 * * r 2001

2001 q 0 r 2002

2002 q 0 r 2002

2002 _ * l 2003

2003 0 1 l 2004

2004 * * l 2004

2004 _ * l 2005

2005 * * l 2005

2005 _ * r q0

;Checksum COMER

3000 0 1 r 3000

3000 1 0 r 3000

3000 _ * * halt

; Suma de dos binarios en aritmética de complemento a 1. Suma intermedia RFC 791

; Cálculo del Checksum de acuerdo con las RFC's del IETF

; Entradas: dos números binarios, separados por un espacio, pe. '100 111'

; Programó: Roberto Cárdenas

; Fecha: febrero 27 de 2022

;Primera parte a la derecha

0 * * l qx

qx * * l qz

qz * * l qp

qp * s r qy

qy * * r qm

qm * * r qq

qq 1 0 r qq

qq 0 1 r qq

qq * * * q0

q0 _ _ r 1

q0 * * r q0

;Segunda parte a la derecha

1 1 0 r 1

1 0 1 r 1

1 * * * q1

;Comienza la suma

q2 _ _ r q1

q2 * * r q2

;

q1 _ _ l 2

q1 * * r q1

;

2 0 _ l 3x

2 1 _ l 3y

2 _ _ l 7

3x _ _ l 4x

3x * * l 3x

3y _ _ l 4y

3y * * l 3y

$4x0xrq2$
 $4x1yrq2$
 $4x_xrq2$
 $4x**l4x$; skip the x/y's
 $4y0l*5$
 $4y10l4y$
 $4y_l*5$
 $4y**l4y$; skip the x/y's
 $5xxl6$
 $5yy l6$
 5_l6
 $5**r5$
 $60xrq2$
 $61yrq2$

$7x0l7$
 $7y1l7$
 7_l10
 $7**l7$

;no hay carry?

10_l99
 $99s_r100$
 99_r150
 150_r150
 1500_r3000
 100_r100
 $10000r3000$
 $10011r3000$

;detectar carry

$10***12$
 $12s_r13$
 13_r8

;SUMAR CARRY

$81or9$
 $91ur9$

9 0 o r 9

;Llegar a separador

9 _ * l 900

900 * * l 900

900 u * * 1001

900 o * * 1020

900 _ * r 2001

;Lee u

1001 u 1 r 1002

;Lee o

1020 o 0 r 1002

;escribe q y se devuelve

1002 * * r 1002

1002 _ * r 1003

1003 * * r 1003

1003 _ q l 1004

1004 * * l 1004

1004 o * * 1020

1004 u * * 1001

1004 _ * l 900

;última vuelta al comienzo

2001 * * r 2001

2001 q 1 r 2002

2002 q 1 r 2002

2002 _ * l 2003

2003 1 0 l 2004

2004 * * l 2004

2004 _ * l 2005

2005 * * l 2005

2005 _ * r q0

;Checksum RFC

3000 1 1 r 3000

3000 0 0 r 3000

3000 _ * * halt

; Suma de dos binarios en aritmética de complemento a 1. Suma final RFC 791

; Cálculo del Checksum de acuerdo a las RFC's del IETF

; Entradas: dos número binarios, separados por un espacio, pe. '100 111'

; Programó: Roberto Cárdenas

; Fecha: febrero 27 de 2022

;Primera parte a la derecha

0 * * l qx

qx * * l qz

qz * * l qp

qp * s r qy

qy * * r qm

qm * * r qq

qq 1 0 r qq

qq 0 1 r qq

qq * * * q0

q0 _ _ r 1

q0 * * r q0

;Segunda parte a la derecha

1 1 0 r 1

1 0 1 r 1

1 * * * q1

;Comienza la suma

q2 _ _ r q1

q2 * * r q2

;

q1 _ _ l 2

q1 * * r q1

;

2 0 _ l 3x

2 1 _ l 3y

2 _ _ l 7

3x _ _ l 4x

3x * * l 3x

3y _ _ l 4y

3y * * l 3y

4x 0 x r q2
 4x 1 y r q2
 4x _ x r q2
 4x * * l 4x ; skip the x/y's
 4y 0 1 * 5
 4y 1 0 l 4y
 4y _ 1 * 5
 4y * * l 4y ; skip the x/y's
 5 x x l 6
 5 y y l 6
 5 _ _ l 6
 5 * * r 5
 6 0 x r q2
 6 1 y r q2

7 x 0 l 7
 7 y 1 l 7
 7 _ _ l 10
 7 * * l 7

;no hay carry?

10 _ * l 99
 99 s _ r 100
 99 _ * r 150
 150 _ * r 150
 150 0 _ r 3000
 100 _ * r 100
 100 0 1 r 3000
 100 1 0 r 3000

;detectar carry

10 * * * 12
 12 s _ r 13
 13 _ _ r 8

;SUMAR CARRY

8 1 o r 9
 9 1 u r 9

9 0 o r 9

;Llegar a separador

9 _ * l 900

900 * * l 900

900 u * * 1001

900 o * * 1020

900 _ * r 2001

;Lee u

1001 u 1 r 1002

;Lee o

1020 o 0 r 1002

;escribe q y se devuelve

1002 * * r 1002

1002 _ * r 1003

1003 * * r 1003

1003 _ q l 1004

1004 * * l 1004

1004 o * * 1020

1004 u * * 1001

1004 _ * l 900

;última vuelta al comienzo

2001 * * r 2001

;2001 q 0 r 2002

2001 q 1 r 2002

2002 q 1 r 2002

2002 _ * l 2003

2003 1 0 l 2004

2004 * * l 2004

2004 _ * l 2005

2005 * * l 2005

2005 _ * r q0

;Checksum RFC

3000 1 0 r 3000

3000 0 1 r 3000

3000 _ * * halt