

TEMA 3 TÉCNICAS DE PROGRAMACIÓN. LISTADOS Y FIGURAS

Ec. de segundo grado. Primer nivel

```
Inicio
  leer a, b, c
  si a ≠ 0 entonces
    disc ←  $b^2 - 4ac$ 
    si disc > 0 entonces
      calcular soluciones reales
    si_no
      si disc = 0 entonces
        calcular solución doble
      si_no
        calcular soluciones complejas
      fin_si
    fin_si
  si_no
    escribir 'No es de segundo grado'
  fin_si
Fin
```

Ec. de segundo grado, segundo nivel

```
Inicio
  leer a, b, c
  si a ≠ 0 entonces
    disc ←  $b^2 - 4ac$ 
    si disc > 0 entonces
      escribir  $(-b + \sqrt{disc})/(2a)$ 
      escribir  $(-b - \sqrt{disc})/(2a)$ 
    si_no
      si disc = 0 entonces
        escribir  $-b/(2a)$  'doble'
      si_no
        Re ←  $-b/(2a)$ 
        Im ←  $\sqrt{-disc}/(2a)$ 
        escribir Re '+' Im 'i'
        escribir Re '-' Im 'i'
      fin_si
    fin_si
  si_no
    escribir 'No es de segundo grado'
  fin_si
Fin
```

MCD, primer nivel

```

Inicio
  leer m, n
  { colocarlos de forma que  $m \geq n$  : }
  si m < n entonces
    intercambiar valores de m y n
  fin si
  r ← m MOD n
  mientras r > 0 hacer
    m ← n
    n ← r
    r ← m MOD n
  fin mientras
  escribir 'El MCD es ' n
Fin

```

Expresar en binario un entero positivo

```

Inicio
  leer n
  si n > 0 entonces
    { obtener maxpot: máxima potencia de 2 que se puede restar a n: }
    maxpot ← 1
    mientras maxpot ≤ n hacer
      maxpot ← 2 * maxpot
    fin mientras
    { V:  $1 \leq \text{maxpot} = \text{potencia de } 2. n < \text{maxpot}$  }
    maxpot ← maxpot / 2
    { V: maxpot es la mayor potencia de 2 que se puede restar de n }
    repetir
      { escribir bit y actualizar n: }
      si n ≥ maxpot entonces
        escribir 1
        n ← n - maxpot
      si_no
        escribir 0
      fin_si
      { rebajar maxpot : }
      maxpot ← maxpot / 2
    hasta_que maxpot = 0
    { V: maxpot = 0 y ya no hay bits por escribir }
  si_no
    escribir 'No previsto'
  fin_si
Fin

```

“Traza para MCD 36 120”

MCD, segundo nivel

```

0 Inicio
1   leer m, n
2   si m < n entonces
3     x ← m
4     m ← n
5     n ← x
6   fin si
7   r ← m MOD n
8   mientras r > 0 hacer
9     m ← n
10    n ← r
11    r ← m MOD n
12  fin mientras
13  escribir 'El MCD es ' n
14 Fin

```

línea	entrada	m	n	x	r	condición	salida
0		?	?	?	?		
1	36 120	36	120	?	?		
2		36	120	?	?	m>n=T	
3		36	120	36	?		
4		120	120	36	?		
5		120	36	36	?		
6							
7		120	36	36	12		
8		120	36	36	12	r>0=T	
9		36	36	36	12		
10		36	12	36	12		
11		36	12	36	0		
12							
13	12;8	36	12	36	0	r>0=F	
14		36	12	36	0		12
14							

ILog2

```

Inicio {Pre: m>0 }
  maxpot ← 1
  mientras maxpot ≤ m hacer
    maxpot ← 2 * maxpot
  fin mientras
  {V: 1 ≤ maxpot = potencia de 2; m < maxpot }
  maxpot ← maxpot / 2
  devolver maxpot
Fin

```

```

Entorno local: Variables: auxiliar : tipo T, para intercambio
Inicio
  auxiliar ← m
  m ← n
  n ← auxiliar
Fin

```