

```

//
Dieses bekannte Problem, das in Syrakus in den Vereinigten Staaten entstanden ist, hat eine Vermutung
hervorgebracht wer wird noch nicht bewiesen.
Nehmen Sie irgendein natürliches Ganzes. Wenn er gerade ist, durch 2 teilt.
Wenn er ungerad ist, mit 3 multipliziert und 1 hinzufügt.
Setzen Sie fort auf die gleiche Art und Weise... und sich anhalten, wenn Sie in 1 ankommen.
Die Vermutung bedeutet, daß man immer in 1 ankommt.
Man kann mit jedem Ganz es eine Höhe verbinden, wer ist die Anzahl der notwendigen Operationen, bevor es in 1
ankommt.
Beispiel:
h(1)=0
h(32)=5 : 32 16 8 4 2 1
h(5)=5 : 5 16 8 4 2 1
h(9)=19 : 9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
//

lerne Syrakus :n
setze "h 0 setze "k :n
solange (:k > 1) [
    prüfe (ganzzahl (:k / 2)) = (:k / 2)
    wennwahr setze "k :k /2
    wennfalsch setze "k (3* :k )+1
    setze "h :h + 1
]
rückgabe :h
Ende

lerne Parität :n
setze "k :n
setze "gerad 0 setze "ungerad 0
solange (:k > 1) [
    prüfe (ganzzahl (:k / 2)) = (:k / 2)
    wennwahr [ setze "k :k /2 setze "gerad :gerad + 1 ]
    wennfalsch [ setze "k (3* :k )+1 setze "ungerad :ungerad +1 ]
]
rückgabe :gerad - :ungerad
Ende

lerne ungerad :n
setze "k :n
setze "ungerad 0
solange (:k > 1) [
    prüfe (ganzzahl (:k / 2)) = (:k / 2)
    wennwahr [ setze "k :k /2 setze "ungerad :ungerad - 1 ]
    wennfalsch [ setze "k (3* :k )+1 setze "ungerad :ungerad +1 ]
]
rückgabe :ungerad
Ende

lerne graph :c :n
Bild löschttext vi fi schwarz
setze "AltesOrt [ -240 (-250) ]
Segment [ -240 (-250) ] [ 250 (-250)]
Segment [ -240 (-250) ] [ -240 (-50)]
zeichnetext [ -265 (-70) 0] "S(n):
zeichnetext [ -265 (-263) 0] :c
zeichnetext [ -250 + :n (-263) 0] :c + :n
setze "sup 0 fi blau setze "inf 1E8
setze "xsup :c setze "xinf :c
setze "sup_Ort [ :c (:c + :n) 0 ] setze "inf_Ort [ :c (:c + :n) 0 ]
wiederhole :n [
    setze "y Syrakus ( :c + whzahl )
    wenn :y >= :sup [
        setze "sup_Ort Satz Satz -250 + whzahl (-250) + :y 0
        setze "xsup whzahl
        setze "sup :y
    ]
    wenn :y <= :inf [
        setze "inf_Ort Satz Satz -250 + whzahl (-250) + :y 0
        setze "xinf whzahl
        setze "inf :y
    ]
]
Segment :AltesOrt [ -240 + whzahl (-250) + :y ]
setze "AltesOrt Satz -240 + whzahl (-250) + :y
]
fi blau

```

```

zeichnetext :sup_Ort Satz Satz Satz [Ssup( ) :xsup( ) =] :sup
zeichnetext :inf_Ort Satz Satz Satz [Sinf( ) :xinf( ) =] :inf
schreiblinie Satz Satz Satz [ minimaler Wert : ] :inf "Amplitude: :sup - :inf
Ende

```

```

lerne Veränderung :c :n
setze "sup 0 setze "mini 1E4
wiederhole :n [
    setze "y Syrakus (:c + whzahl )
    wenn :y >= :sup setze "sup :y
    wenn :y <= :mini setze "mini :y
]
rückgabe ganzzahl ((:sup - :mini) / :n )
Ende

```

```

lerne graph2 :c :n
//Bild löschttext// vi fi schwarz
zeichnetext [ -51 56 0 ] [ _Syrakus Schwankungen: ] // _Syrakus (mit einem Gedankenstrich
vorne), um nicht mit dem Syrakus Prozedur
zu verwechseln. //

```

```

setze "AltesOrt [ -240 (-20) ]
Segment [ -240 (-20) ] [ 250 (-20)]
Segment [ -240 (-20) ] [ -240 73]
zeichnetext [ -265 73 0 ] "Delta(n):
zeichnetext [ -265 (-25) 0 ] :c
zeichnetext [ -250 + :n (-25) 0 ] :c + :n
setze "sup 0 setze "inf 1E5
setze "xsup :c setze "xinf :c
setze "sup_Ort [ :c (:c + :n) 0 ] setze "inf_Ort [ :c (:c + :n) 0 ]
fi rot
wiederhole :n [
    setze "y Veränderung ( :c + whzahl ) 10
    Segment :AltesOrt [ -240 + whzahl (-20) + :y ]
    setze "AltesOrt Satz -240 + whzahl (-20) + :y
    wenn :y >= :sup [
        setze "sup_Ort Satz Satz -255 + whzahl (20) + :y 0
        setze "xsup whzahl
        setze "sup :y
    ]
    wenn :y <= :inf [
        setze "inf_Ort Satz Satz -255 + whzahl (20) + :y 0
        setze "xinf whzahl
        setze "inf :y
    ]
]
zeichnetext :sup_Ort Satz Satz Satz [Dsup( ) :xsup( ) =] :sup
zeichnetext :inf_Ort Satz Satz Satz [Dinf( ) :xinf( ) =] :inf
Ende

```

```

lerne graph3 :c :n
//Bild löschttext// vi fi schwarz
zeichnetext [ -51 256 0 ] [ Studie der Parität: ]
setze "AltesOrt [ -240 191]
Segment [ -240 191 ] [ 250 191]
Segment [ -240 147 ] [ -240 241]
zeichnetext [ 13 235 0 ] "Gerad(n):
zeichnetext [ -265 185 0 ] :c
zeichnetext [ -250 + :n 185 0 ] :c + :n
setze "sup 0 setze "inf 1E5
setze "xsup :c setze "xinf :c
setze "sup_Ort [ :c (:c + :n) 0 ] setze "inf_Ort [ :c (:c + :n) 0 ]
fi braun
wiederhole :n [
    setze "y Parität ( :c + whzahl )
    Segment :AltesOrt [ -240 + whzahl 191+ :y ]
    setze "AltesOrt Satz -240 + whzahl 191+ :y
    wenn :y >= :sup [
        setze "sup_Ort Satz Satz -255 + whzahl 191+ :y 0
        setze "xsup whzahl
        setze "sup :y
    ]
    wenn :y <= :inf [
        setze "inf_Ort Satz Satz -255 + whzahl 191+ :y 0
        setze "xinf whzahl
        setze "inf :y
    ]
]

```

```

]
zeichnetext :sup_Ort Satz Satz Satz [Psup( ) :xsup ( ) =] :sup
zeichnetext :inf_Ort Satz Satz Satz [Pinf( ) :xinf ( ) =] :inf
setze "AltesOrt [ -240 191]
zeichnetext [ 13 145 0] "Imgerad(n):
setze "sup -1E5 setze "inf 0
setze "xsup :c setze "xinf :c
setze "sup_Ort [ :c (:c + :n) 0 ] setze "inf_Ort [ :c (:c + :n) 0 ]
fi rosa
wiederhole :n [
  setze "y ungerad ( :c + whzahl )
  Segment :AltesOrt [ -240 + whzahl 191 + :y ]
  setze "AltesOrt Satz -240 + whzahl 191 + :y
  wenn :y >= :sup [
    setze "sup_Ort Satz Satz -255 + whzahl 191 - :y 0
    setze "xsup whzahl
    setze "sup :y
  ]
  wenn :y <= :inf [
    setze "inf_Ort Satz Satz -255 + whzahl 191+ :y 0
    setze "xinf whzahl
    setze "inf :y
  ]
]
zeichnetext :sup_Ort Satz Satz Satz [ sup_ungerad( ) :xsup ( ) =] :sup
zeichnetext :inf_Ort Satz Satz Satz [ inf_ungerad( ) :xinf ( ) =] :inf
Ende

lerne zeichne
graph 10 500 graph2 10 500 graph3 10 500
Ende

rollauf zeichne

// Gemäß Auftrag (Arbeitsfenster) tippen:

zeichne

//

```