```
'{$STAMP BS2pe}
'{$PBASIC 2.5}
' program cordic-atn.bpe, version 12/2005
' Tracy Allen, eme systems, http://www.emesystems.com
' CORDIC computation of ARCTAN from values x and y, computes to 2 decimal places
' version 12/2005
'    improves accuracy for small values of x and y by use of a scaling wrapper
'    corrects error in final angle calculation in quadrant23
'    uses 16200 for 45 degrees instead of 16384, to better handle x=0 issues.
'   also see: http://forums.parallax.com/forums/default.aspx?f=5&p=1&m=102008
' The CORDIC algorithm works for inputs in quadrants 1 and 4, so
'   this program provides a wrapper for quadrants 2 and 3.
' The anglular measure in this program assigns the value 16200 to a 45 degree angle,
'   and that is used to calculate the values in the table, tans.
' A step at the end of the program (**18204) converts this to standard degrees.
' Wrapper scales small numbers up (without change of ratio y/x) and scales back down at end.
' This program provides three different methods for computing the rotation (#DEFINE method=).
' One uses straihtforward IF-THEN branching on the sign, while the other
' two eschew the IF-THEN in favor of algebraic manipulation of the sign.
' The trick for the Stamp is division of a negative number.
' Examples:
'   45 degrees, x=10000, y=10000
'   22.5 degrees, x=9239, y=3827
'   30 degrees, x=8660, y=5000
' The program also computes the length of the vector SQR(x^2+y^2),
' Enable the DEBUG statement within the CORDIC loop to see inner operation.

' table of atn(1), atn(1/2), atn(1/4),... scaled for atn(1)=16200
atans DATA WORD 16200, WORD 9563, WORD 5053, WORD 2565
        DATA WORD 1287, WORD 644, WORD 322, WORD 161'
        DATA WORD 81, WORD 40, WORD 20, WORD 10
        DATA WORD 5, WORD 3, WORD 1, WORD 1, WORD 0

Cgain CON 39797  ' 0.607252935 * 65536 CORDIC gain

x VAR WORD              ' entered by user
xs VAR x.BIT15  ' sign bit
xx VAR WORD     ' for calculation
y VAR WORD              ' entered by user
ys VAR y.BIT15  ' sign bit
yy VAR WORD     ' for calculation
z VAR WORD              ' angle accumulator
zs VAR z.BIT15  ' sign bit
exp VAR NIB     ' for scaling the input numbers, wrapper

atan2 VAR WORD  ' to hold values from the table of atans.
idx VAR NIB     ' index for the main calculation loop
quadrant23 VAR BIT      ' to hold the quadrant flag, wrapper


top:
    DEBUG CR,"enter signed numbers -23172 to +23172",CR
DO
  DEBUG "enter x coordinate:"
  DEBUGIN SDEC x
  DEBUG "enter y coordinate:"
  DEBUGIN SDEC y

 ' adjust the numbers, if small make them larger
 ' algorithm: multiply both x and y by a factor that makes the largest one
 ' between 8192 and 16384
 ' then preapply the CORDIC gain multiplier 39797
```

```
  '
   exp = 14 - ((NCD ABS x) MIN (NCD ABS y) MAX 14)
   x = -xs ^ (ABS x << exp ** Cgain) + xs
   y = -ys ^ (ABS y << exp ** Cgain) + ys

   quadrant23=xs
   IF quadrant23 THEN x=-x    ' move to 1st or 4th quadrant, where cordic works

   z=0                    ' initialize the angle accumulator
   FOR idx=0 TO 15
   READ idx*2+atans, WORD atan2

 #DEFINE method=3  ' method 1 uses branch on sign, method 2&3 use computed direction

 #SELECT method
  #CASE 1
   IF ys THEN
     ' y<0, ys=1, d=+1
       xx = x +(ABS y >> idx)  ' y<0 on entry
       y = y + (-xs ^ (ABS x >> idx) + xs)
       z = z - atan2
     ELSE
     ' y>=0, ys=0, d=-1
       xx = x + (y >> idx)      ' y>=0 on entry
       y = y - (-xs ^ (ABS x >> idx) + xs)
       z = z + atan2
   ENDIF

 #CASE 2
    ' term (-ys ^ (ABS y >> idx) + ys) is division by power of 2 sign extended
    ' term -ys ^ (.) + ys is 2s complement, addition/subtraction depending on sign bit, ys.
   xx = x + (-ys ^ (-ys ^ (ABS y >> idx) + ys) + ys)
   yy = y - (-ys ^ (-xs ^ (ABS x >> idx) + xs) + ys)
   z = z + (-ys ^ atan2 + ys)
   y=yy

 #CASE 3
   ' term (y >> idx | (-ys << (16-idx))) is division by power of 2 sign extended
   xx = x + (-ys ^ (y >> idx | (-ys << (16-idx))) + ys)
   yy = y - (-ys ^ (x >> idx | (-xs << (16-idx))) + ys)
   z = z + (-ys ^ atan2 + ys)
   y=yy
 #ENDSELECT

   x=xx
   ' include the following DEBUG to see internal operation
    ' DEBUG CR,DEC idx,TAB,SDEC atan2,TAB,SDEC x,TAB,SDEC y,TAB,SDEC z
   NEXT

   ' convert to degree measure from internal measure
   z = -zs ^ (ABS z ** 18204) + zs
   ' unwrap for 2nd and 3rd quadrants
   IF quadrant23 THEN
     IF zs THEN z=-18000-z ELSE z=18000-z
   ENDIF

  ' unwrap scale factor to the vector length, sign extended
   x = (x >> exp)|(-xs << (16-exp))

   DEBUG CR,"the angle is: "
   DEBUG REP "-"\zs,DEC ABS z/100, ".",DEC2 ABS z," degrees",TAB
   DEBUG "length: ",DEC x,CR,CR
   LOOP
```