Option Explicit

Public Function ToDegree(Angle As Single) As Single
ToDegree = Angle / 3.14159 * 180#
End Function

Public Function GetAngle(dx As Single, dy As Single) As Single
If dx >= 0# Then
    If dy > 0# Then
        GetAngle = Atn(dx / dy)
    Else
        GetAngle = 3.14159 - Atn(Abs(dx) / Abs(dy))
    End If
Else
    If dy <= 0# Then
        GetAngle = 3.14159 + Atn(Abs(dx) / Abs(dy))
    Else
        GetAngle = 2 * 3.14159 - Atn(Abs(dx) / Abs(dy))
    End If
End If
End Function

Public Sub DegMinSec(decang As Single, degang As Single, minang As Single, secang As Single)
    degang = Int(decang)
    decang = decang - degang
    minang = Int(decang * 60#)
    decang = decang * 60# - minang
    secang = decang * 60#
End Sub

Public Function ToRad(Angle As Single) As Single
ToRad = Angle / 180 * 3.14159
End Function

Public Sub Traverse()
    Dim Azimuth(1 To 6) As Single, Distance(1 To 6) As Single
    Dim Index As Integer
    Dim X0 As Single, Y0 As Single
    Dim x(1 To 6) As Single, y(1 To 6) As Single
    Dim deltax(1 To 6) As Single, deltay(1 To 6) As Single
    Dim Length(1 To 6), Perimeter As Single, Angle(1 To 6) As Single
    Dim AngleDeg(1 To 6) As Single
    Dim AngleMin(1 To 6) As Single
    Dim AngleSec(1 To 6) As Single
    Dim Bearing(1 To 6) As Single
    Dim NorthSouth(1 To 6) As String, EastWest(1 To 6) As String
    X0 = 1000#
    Y0 = 1000#
    For Index = 1 To 6
        Azimuth(Index) = ActiveCell.Offset(Index - 1, 0).Value
        Distance(Index) = ActiveCell.Offset(Index - 1, 1).Value
        x(Index) = Distance(Index) * Sin(ToRad(Azimuth(Index))) + X0
        y(Index) = Distance(Index) * Cos(ToRad(Azimuth(Index))) + Y0
        ActiveCell.Offset(Index - 1, 2).Value = x(Index)
        ActiveCell.Offset(Index - 1, 3).Value = y(Index)
    Next Index
    Perimeter = 0#
    For Index = 1 To 5
        deltax(Index) = x(Index + 1) - x(Index)
        deltay(Index) = y(Index + 1) - y(Index)
        Length(Index) = (deltax(Index) ^ 2 + deltay(Index) ^ 2) ^ (1 / 2)
        Perimeter = Perimeter + Length(Index)
    Next Index
    deltax(6) = x(1) - x(6)
    deltay(6) = y(1) - y(6)
    Length(6) = (deltax(6) ^ 2 + deltay(6) ^ 2) ^ (1 / 2)
    Perimeter = Perimeter + Length(6)
    Range("A1").Value = Perimeter
    For Index = 1 To 6
        ActiveCell.Offset(Index - 1, 5).Value = Length(Index)
        Angle(Index) = ToDegree(GetAngle(deltax(Index), deltay(Index)))
        Call AzimuthToBearing(Angle(Index), Bearing(Index), NorthSouth(Index), EastWest(Index))
    Next Index
End Sub
Module1 - 2

Call DegMinSec(Bearing(Index), AngleDeg(Index), AngleMin(Index), AngleSec(Index))
ActiveCell.Offset(Index - 1, 6).Value = NorthSouth(Index)
ActiveCell.Offset(Index - 1, 7).Value = AngleDeg(Index)
ActiveCell.Offset(Index - 1, 8).Value = AngleMin(Index)
ActiveCell.Offset(Index - 1, 9).Value = AngleSec(Index)
ActiveCell.Offset(Index - 1, 10).Value = EastWest(Index)
Next Index
End Sub

Public Sub AzimuthToBearing(Az As Single, Br As Single, NS As String, EW As String)
    Select Case Az
        Case Is = 0#
            NS = "N"
            EW = "" 
            Br = 0#
        Case Is < 90#
            NS = "N"
            EW = "E"
            Br = Az
        Case Is = 90#
            NS = "" 
            EW = "E"
            Br = 0#
        Case Is < 180#
            NS = "S"
            EW = "E"
            Br = 180# - Az
        Case Is = 180#
            NS = "S"
            EW = "" 
            Br = 0#
        Case Is < 270#
            NS = "S"
            EW = "W"
            Br = Az - 180#
        Case Is = 270#
            NS = "" 
            EW = "W"
            Br = 0#
        Case Is < 360#
            NS = "N"
            EW = "W"
            Br = 360# - Az
        Case Else
            NS = "N"
            EW = "" 
            Br = 0#
    End Select
End Sub