

Option Explicit

'Prozedur zur Erstellung eines Formblatts

Sub Brinell_Formblatt()

'Tabelle löschen

Worksheets("Brinell").Activate

Worksheets("Brinell").Cells.Clear

'Tabelle beschriften

Range("A1").Value = "Kugelradius"

Range("A2").Value = "Abdruckradius"

Range("A3").Value = "Eindringtiefe"

Range("C1").Value = "mm"

Range("C2").Value = "mm"

Range("C3").Value = "mm"

Range("A1").ColumnWidth = 20

Range("B1").ColumnWidth = 10

Range("C1").ColumnWidth = 5

Columns("B").Select

Selection.NumberFormat = "0.000"

Range("B1").Select

End Sub

'Prozedur zur Berechnung der Eindringtiefe

Sub Brinell_Auswertung()

Dim r, x, h As Double

'Eingabewerte lesen

r = Cells(1, 2)

x = Cells(2, 2)

'Berechnung

h = r - Sqr(r * r - x * x)

'Ausgabe

Cells(3, 2) = h

Range("B1").Select

End Sub

Option Explicit

```

Sub Formular()
Worksheets("Kubische Gleichung").Cells.Clear
Cells(3, 1) = "Gleichung:"
Cells(3, 2) = "x" & ChrW(179) & " + "
Range("C3").Interior.ColorIndex = 15
Cells(3, 4) = "x" & ChrW(178) & " + "
Range("E3").Interior.ColorIndex = 15
Cells(3, 6) = "x" & " + "
Range("G3").Interior.ColorIndex = 15
Cells(3, 8) = "="
Cells(3, 9) = "0"
Cells(3, 3).Activate

Cells(5, 1) = "D="
Cells(6, 1) = "x1="
Cells(7, 1) = "x2="
Cells(8, 1) = "x3="
Range("B5:C5").MergeCells = True
Range("B6:C6").MergeCells = True
Range("B7:C7").MergeCells = True
Range("B8:C8").MergeCells = True
End Sub

```

```

Private Sub Worksheet_Change(ByVal Target As Range)
If Target.Row = 3 Then
Select Case Target.Column
Case 3
Cells(3, 5).Activate
Case 5
Cells(3, 7).Activate
Case 7
Call Auswertung
Cells(3, 3).Activate
End Select
End If
End Sub

```

```

Sub Auswertung()
Dim a, b, c, D, p, q, r, s, u, v, w, z As Double
Dim y1, y2, y3, y2i, y3i, x1, x2, x3 As Double

a = Cells(3, 3)
b = Cells(3, 5)
c = Cells(3, 7)

p = b - (a * a) / 3
q = (2 * a * a * a) / 27 - (a * b) / 3 + c
D = (q / 2) ^ 2 + (p / 3) ^ 3
Cells(5, 2) = D
If D < 0 Then
r = Sqr(-(p / 3) ^ 3)
z = -(q / 2) / r
w = Atn(-z / Sqr(-z * z + 1)) + 2 * Atn(1)
y1 = 2 * r ^ (1 / 3) * Cos(w / 3)
y2 = 2 * r ^ (1 / 3) * Cos(w / 3 + 8 * Atn(1) / 3)
y3 = 2 * r ^ (1 / 3) * Cos(w / 3 + 16 * Atn(1) / 3)
x1 = y1 - a / 3
x2 = y2 - a / 3
x3 = y3 - a / 3
Cells(6, 2) = x1
Cells(7, 2) = x2
Cells(8, 2) = x3
Cells(7, 4) = ""
Cells(8, 4) = ""
Cells(7, 5) = ""
Cells(8, 5) = ""
ElseIf D = 0 Then
y1 = 2 * (-q / 2) ^ (1 / 3)
y2 = (-q / 2) ^ (1 / 3)
y3 = y2
x1 = y1 - a / 3
x2 = y2 - a / 3
x3 = y3 - a / 3
Cells(6, 2) = x1

```

```
Cells(7, 2) = x2
Cells(8, 2) = x3
Cells(7, 4) = ""
Cells(8, 4) = ""
Cells(7, 5) = ""
Cells(8, 5) = ""
Else
  s = Sqr(D)
  z = -q / 2 + s
  If z >= 0 Then
    u = z ^ (1 / 3)
  Else
    u = -(-z) ^ (1 / 3)
  End If
  z = -q / 2 - s
  If z >= 0 Then
    v = z ^ (1 / 3)
  Else
    v = -(-z) ^ (1 / 3)
  End If
  y1 = u + v
  y2 = -(u + v) / 2
  y2i = Sqr(3) * (u - v) / 2
  y3 = -(u + v) / 2
  y3i = -Sqr(3) * (u - v) / 2
  x1 = y1 - a / 3
  x2 = y2 - a / 3
  x3 = y3 - a / 3
  Cells(6, 2) = x1
  Cells(7, 2) = x2
  Cells(8, 2) = x3
  Cells(7, 4) = "+ i"
  Cells(8, 4) = "+ i"
  Cells(7, 5) = y2i
  Cells(8, 5) = y3i
End If
End Sub
```

Option Explicit

'Prozedur zur Erstellung eines Formblatts

Private Sub Minimum_Formblatt()

'Tabelle löschen

Worksheets("Minimum").Activate
Worksheets("Minimum").Cells.Clear

'Tabelle beschriften

Range("A1").Value = "Volumen V ="
Range("A2").Value = "Startwert d1 ="
Range("A3").Value = "Startwert d2 ="
Range("A4").Value = "Abschaltgrenze"
Range("C1").Value = "cm" + ChrW(179)
Range("C2").Value = "mm"
Range("C3").Value = "mm"
Range("E1").Value = "Oberfläche [cm^2]"
Range("F1").Value = "O'(d)"
Range("G1").Value = "O''(d)]"
Range("B6").Value = "d [cm]"Range("A1").ColumnWidth = 15
Range("B1").ColumnWidth = 10
Range("C1").ColumnWidth = 5
Range("D1").ColumnWidth = 1
Range("E1").ColumnWidth = 15
Range("F1").ColumnWidth = 15
Range("G1").ColumnWidth = 15
Columns("B").Select
Selection.NumberFormat = "0.000"
Range("B1").Select

End Sub

Private Sub Minimum_Testdaten()

Cells(1, 2) = 50
Cells(2, 2) = 10
Cells(3, 2) = 100
Cells(4, 2) = 0.01

End Sub

Private Function Ob(V, d) 'Oberfläche in cm^3

Dim r As Double
Dim pi As Double
pi = 4 * Atn(1) 'Konstante pi
r = d / 2 'Radius in cm
Ob = 2 * pi * r * r + 2 * V / r

End Function

Private Function Ob1(V, d) '1. Ableitung der Oberfläche

Dim r As Double
Dim pi As Double
pi = 4 * Atn(1) 'Konstante pi
r = d / 2 'Radius in cm
Ob1 = 4 * pi * r - 2 * V / (r * r)

End Function

Private Function Ob2(V, d) '2. Ableitung der Oberfläche

Dim r As Double
Dim pi As Double
pi = 4 * Atn(1) 'Konstante pi
r = d / 2 'Radius in cm
Ob2 = 4 * pi + 4 * V / (r * r * r)

End Function

Private Sub Minimum_Auswertung()

Dim d, d1, d2, V, e As Double
Dim i As Integer

'Eingabewerte lesen

V = Cells(1, 2) 'Volumen in cm^3
d1 = Cells(2, 2) / 10 'Startwert 1 in cm
d2 = Cells(3, 2) / 10 'Startwert 2 in cm
e = Cells(4, 2) 'Abschaltkriterium

'Startüberprüfung

If Ob1(V, d1) > 0 Then

```
If Ob1(V, d2) <= 0 Then
    'Startwerte korrekt
Else
    MsgBox "Startwerte falsch!", _
        vbInformation & vbOKOnly
    Exit Sub
End If
Else
    If Ob1(V, d2) > 0 Then
        d = d1: d1 = d2: d2 = d
    Else
        MsgBox "Startwerte falsch!", _
            vbInformation & vbOKOnly
        Exit Sub
    End If
End If

'Berechnung
Cells(2, 5) = Ob(V, d1)
Cells(2, 6) = Ob1(V, d1)
Cells(2, 7) = Ob2(V, d1)
Cells(3, 5) = Ob(V, d2)
Cells(3, 6) = Ob1(V, d2)
Cells(3, 7) = Ob2(V, d2)

i = 6
Do
    d = d1 - Ob1(V, d1) * (d2 - d1) _
        / (Ob1(V, d2) - Ob1(V, d1))
    If Ob1(V, d) > 0 Then
        d1 = d
    Else
        d2 = d
    End If
    i = i + 1
    Cells(i, 2) = d
    Cells(i, 5) = Ob(V, d)
    Cells(i, 6) = Ob1(V, d)
    Cells(i, 7) = Ob2(V, d)
'Abbruchkriterium
Loop While Abs(Ob1(V, d)) > e

End Sub
```

Option Explicit

'Prozedur zur Erstellung eines Formblatts

Private Sub Maximum_Formblatt()

'Tabelle löschen

Worksheets("Maximum").Activate
Worksheets("Maximum").Cells.Clear

'Tabelle beschriften

Range("A1").Value = "Kantenlänge a ="
Range("A2").Value = "Startwert x ="
Range("A3").Value = "Abschaltgrenze"
Range("C1").Value = "cm"
Range("C2").Value = "cm"
Range("E1").Value = "Volumen [cm^3]"
Range("F1").Value = "V'(x)"
Range("G1").Value = "V''(x)"]"
Range("B6").Value = "x [cm]"
Range("A1").ColumnWidth = 15
Range("B1").ColumnWidth = 10
Range("C1").ColumnWidth = 5
Range("D1").ColumnWidth = 1
Range("E1").ColumnWidth = 15
Range("F1").ColumnWidth = 15
Range("G1").ColumnWidth = 15
Columns("B").Select
Selection.NumberFormat = "0.000"
Range("B1").Select

End Sub

Private Sub Maximum_Testdaten()

Cells(1, 2) = 50
Cells(2, 2) = 2
Cells(3, 2) = 0.01

End Sub

Private Function Vol(a, x) 'Volumen in cm^3

Vol = x * (a - 2 * x) ^ 2

End Function

Private Function Vol(a, x) '1. Ableitung des Volumens

Vol = 12 * x * x - 8 * a * x + a * a

End Function

Private Function Vo2(a, x) '2. Ableitung des Volumens

Vo2 = 24 * x - 8 * a

End Function

Private Sub Maximum_Auswertung()

Dim a, x, e As Double
Dim i As Integer

'Eingabewerte lesen

a = Cells(1, 2) 'Kantenlänge in cm
x = Cells(2, 2) 'Startwert in cm
e = Cells(3, 2) 'Abschaltkriterium

'Berechnung

Cells(2, 5) = Vol(a, x)
Cells(2, 6) = Vol(a, x)
Cells(2, 7) = Vo2(a, x)

i = 6

Do

x = x - Vol(a, x) / Vo2(a, x)

i = i + 1

Cells(i, 2) = x

Cells(i, 5) = Vol(a, x)

Cells(i, 6) = Vol(a, x)

Cells(i, 7) = Vo2(a, x)

'Abbruchkriterium

Loop While Abs(Vol(a, x)) > e

End Sub

Option Explicit

'Gauss-Elimination

Private Sub Werte(n, A, x)

Dim Shp As Shape

Dim i, j As Integer

'Tabelle löschen

Worksheets("Temperaturverteilung").Activate

Worksheets("Temperaturverteilung").Cells.Clear

For Each Shp In Shapes

Shp.Delete

Next

'Werte

Range("A1:P15").ColumnWidth = 5

Range("A20:G24").Select

Selection.NumberFormat = "0.00"

For i = 1 To n

For j = 1 To n + 1

Cells(i, j) = 0

If i = j Then Cells(i, j) = 4

Next j

Next i

Cells(1, 2) = -1: Cells(1, 6) = -1

Cells(2, 1) = -1: Cells(2, 3) = -1

Cells(2, 7) = -1

Cells(3, 2) = -1: Cells(3, 4) = -1

Cells(3, 8) = -1

Cells(4, 3) = -1: Cells(4, 5) = -1

Cells(4, 9) = -1

Cells(5, 4) = -1: Cells(5, 10) = -1

Cells(6, 1) = -1: Cells(6, 7) = -1

Cells(6, 11) = -1

Cells(7, 2) = -1: Cells(7, 6) = -1

Cells(7, 8) = -1: Cells(7, 12) = -1

Cells(8, 3) = -1: Cells(8, 7) = -1

Cells(8, 9) = -1: Cells(8, 13) = -1

Cells(9, 4) = -1: Cells(9, 8) = -1

Cells(9, 10) = -1: Cells(9, 14) = -1

Cells(10, 5) = -1: Cells(10, 9) = -1

Cells(10, 15) = -1

Cells(11, 6) = -1: Cells(11, 12) = -1

Cells(12, 7) = -1: Cells(12, 11) = -1

Cells(12, 13) = -1

Cells(13, 8) = -1: Cells(13, 12) = -1

Cells(13, 14) = -1

Cells(14, 9) = -1: Cells(14, 13) = -1

Cells(14, 15) = -1

Cells(15, 10) = -1: Cells(15, 14) = -1

Cells(1, 16) = 24

Cells(2, 16) = 18

Cells(3, 16) = 24

Cells(4, 16) = 36

Cells(5, 16) = 84

Cells(6, 16) = 22

Cells(7, 16) = 0

Cells(8, 16) = 0

Cells(9, 16) = 0

Cells(10, 16) = 30

Cells(11, 16) = 34

Cells(12, 16) = 24

Cells(13, 16) = 32

Cells(14, 16) = 24

Cells(15, 16) = 32

End Sub

Private Sub Werte_Lesen(n, A, x)

Dim i, j As Integer

'Bestimmung belegter Zeilen

'und Definition der notwendigen Datenfelder

Cells(Rows.Count, 1).End(xlUp).Select

n = ActiveCell.Row

ReDim A(n, n + 1), x(n) As Double

```

For i = 1 To n
  For j = 1 To n + 1
    A(i, j) = Cells(i, j)
  Next j
Next i
End Sub

Private Sub Subtrahiere_Gleichung(A, n, i, j)
  Dim k As Integer
  Dim c As Double

  c = A(j, i) / A(i, i)
  For k = i + 1 To n + 1
    A(j, k) = A(j, k) - c * A(i, k)
  Next k
End Sub

Private Function Summe(A, x, n, i)
  Dim s As Double
  Dim k As Integer

  s = A(i, n + 1)
  For k = i + 1 To n
    s = s - A(i, k) * x(k)
  Next k
  Summe = s
End Function

Sub Auswertung()
  ReDim A(1, 1), x(1) As Double
  Dim n, i, j, k As Integer

  n = 15
  ReDim A(n, n + 1), x(n) As Double
  Call Werte(n, A, x)
  Call Werte_Lesen(n, A, x)
  For i = 1 To n - 1
    For j = i + 1 To n
      Call Subtrahiere_Gleichung(A, n, i, j)
    Next j
  Next i

  For i = n To 1 Step -1
    x(i) = Summe(A, x, n, i) / A(i, i)
    j = Int((i - 1) / 5) + 1
    k = i - (j - 1) * 5
    Cells(j + 20, k + 1) = x(i)
    Cells(i, n + 3) = x(i)
  Next i

  Cells(20, 1) = 12
  Cells(20, 2) = 12
  Cells(20, 3) = 18
  Cells(20, 4) = 24
  Cells(20, 5) = 36
  Cells(20, 6) = 42
  Cells(20, 7) = 42
  Cells(21, 7) = 42
  Cells(22, 7) = 30
  Cells(23, 7) = 16
  Cells(24, 7) = 16
  Cells(24, 6) = 16
  Cells(24, 5) = 24
  Cells(24, 4) = 32
  Cells(24, 3) = 24
  Cells(24, 2) = 16
  Cells(24, 1) = 17
  Cells(23, 1) = 18
  Cells(22, 1) = 22
  Cells(21, 1) = 12

  Range("A20:G24").Select
  Charts.Add
  ActiveChart.ChartType = xlSurface
  ActiveChart.SetSourceData _
    Source:=Sheets("Temperaturverteilung").Range( _
      "A20:G24"), PlotBy:=xlRows
  ActiveChart.Location Where:=xlLocationAsObject, Name:= _

```



```
"Temperaturverteilung"  
With ActiveChart  
    .HasTitle = True  
    .ChartTitle.Characters.Text = "Temperaturverteilung"  
    .Axes(xlCategory).HasTitle = False  
    .Axes(xlSeries).HasTitle = False  
    .Axes(xlValue).HasTitle = False  
End With  
ActiveChart.Corners.Select  
With ActiveChart  
    .Elevation = 25  
    .Rotation = 211  
End With  
End Sub
```

```
Option Explicit
```

```
Private Sub Testdaten()
```

```
    Dim i, j As Integer
```

```
'Tabelle löschen
```

```
    Worksheets("Simplex").Activate
```

```
    Worksheets("Simplex").Cells.Clear
```

```
'Werte
```

```
    Range("A1:C3").ColumnWidth = 10
```

```
    Range("A1:B1").Select
```

```
    Selection.NumberFormat = "#0"
```

```
    Range("A3:C5").Select
```

```
    Selection.NumberFormat = "0.000"
```

```
    Cells(1, 1) = 2
```

```
    Cells(1, 2) = 2
```

```
    Cells(3, 1) = 6
```

```
    Cells(3, 2) = 4
```

```
    Cells(3, 3) = 160
```

```
    Cells(4, 1) = 2
```

```
    Cells(4, 2) = 4
```

```
    Cells(4, 3) = 120
```

```
    Cells(5, 1) = -3
```

```
    Cells(5, 2) = -5
```

```
    Cells(5, 3) = 0
```

```
End Sub
```

```
Private Sub Auswertung()
```

```
    Dim m, n, i, j, k, pz, ps, min, z As Integer
```

```
    Dim d As Double
```

```
    ReDim A(1, 1), B(1, 1) As Double
```

```
    m = Cells(1, 1)
```

```
    n = Cells(1, 2)
```

```
    ReDim A(m + 1, n + 1), B(m + 1, n + 1)
```

```
'Daten lesen
```

```
    For i = 1 To m + 1
```

```
        For j = 1 To n + 1
```

```
            A(i, j) = Cells(i + 2, j)
```

```
        Next j
```

```
    Next i
```

```
'Marken setzen
```

```
    z = 0
```

```
    For i = 1 To n
```

```
        z = z + 1
```

```
        Cells(m + 5, i) = z
```

```
    Next i
```

```
    For i = 1 To m
```

```
        z = z + 1
```

```
        Cells(i + 2, n + 3) = z
```

```
    Next i
```

```
'Auswertungsschleife
```

```
    Do
```

```
'Bestimme PivotSpalte mit größtem negativen Wert
```

```
'in der untersten Zeile
```

```
    ps = 0
```

```
    For i = 1 To n
```

```
        If ps = 0 Or A(m + 1, i) < min Then
```

```
            ps = i
```

```
            min = A(m + 1, i)
```

```
        End If
```

```
    Next i
```

```
'Bestimme PivotZeile durch Division der Werte
```

```
'der letzten Spalte durch Elemente in der PivotSpalte,
```

```
'vorausgesetzt dieser Wert ist > 0
```

```
    pz = 0
```

```
    For i = 1 To m
```

```
        If A(i, ps) > 0 Then
```

```

    d = A(i, n + 1) / A(i, ps)
    If pz = 0 Or d < min Then
        min = d
        pz = i
    End If
End If
Next i

```

'Umformung

```

For i = 1 To m + 1
    For j = 1 To n + 1
        If Not i = pz And _
            Not j = ps Then
            B(i, j) = A(i, j) - A(i, ps) * _
                A(pz, j) / A(pz, ps)
        End If
    Next j
Next i
For i = 1 To n + 1
    If Not i = ps Then
        B(pz, i) = A(pz, i) / A(pz, ps)
    End If
Next i
For i = 1 To m + 1
    If Not i = pz Then
        B(i, ps) = -A(i, ps) / A(pz, ps)
    End If
Next i
B(pz, ps) = 1 / A(pz, ps)

```

'Neue Tabelle eintragen

```

For i = 1 To m + 1
    For j = 1 To n + 1
        A(i, j) = B(i, j)
        Cells(i + 2, j) = B(i, j)
    Next j
Next i

```

'Marken vertauschen

```

z = Cells(pz + 2, n + 3)
Cells(pz + 2, n + 3) = Cells(m + 5, ps)
Cells(m + 5, ps) = z

```

'Schleifenbedingung

```

k = 0
For i = 1 To n + 1
    If A(m + 1, i) < 0 Then k = 1
Next i
Loop While k = 1

```

End Sub

```
Option Explicit
```

```
Sub Newton_Leer()
    Call Verlauf_Entfernen
    ThisWorkbook.Worksheets("Newton").Cells.Clear
End Sub
```

```
Sub Newton_Testdaten()
    Call Newton_Leer
    Cells(1, 1) = 0: Cells(1, 2) = 30
    Cells(2, 1) = 10: Cells(2, 2) = 18
    Cells(3, 1) = 20: Cells(3, 2) = 11.5
    Cells(4, 1) = 30: Cells(4, 2) = 10
    Cells(5, 1) = 35: Cells(5, 2) = 10.5
    Cells(6, 1) = 40: Cells(6, 2) = 12.5
    Cells(7, 1) = 50: Cells(7, 2) = 20
End Sub
```

```
Private Sub Werte_Lesen(n, A)
    Dim i, j As Integer
    ,
    'Bestimmung belegter Zeilen
    'und Definition der notwendigen Datenfelder
    Cells(Rows.Count, 1).End(xlUp).Select
    n = ActiveCell.Row
    ReDim A(n, n) As Double
    For i = 1 To n
        For j = 1 To 2
            A(i, j) = Cells(i, j)
        Next j
    Next i
End Sub
```

```
Private Sub Steigungen(n, A)
    Dim i, j As Integer

    For j = 3 To n + 1
        For i = 1 To n - j + 2
            Cells(i, j) = (Cells(i + 1, j - 1) - _
                Cells(i, j - 1)) / _
                (Cells(i + j - 2, 1) - Cells(i, 1))
        Next i
    Next j
    Exit Sub
End Sub
```

```
Private Sub Verlauf(n, A)
    Dim v, b, x, y, z As Double
    Dim i, j, k, p As Integer

    p = 0
    v = Cells(1, 1)
    b = Cells(n, 1)
    For x = v To b Step 1
        y = Cells(1, 2)
        For j = 3 To n + 1
            z = Cells(1, j)
            For k = 1 To j - 2
                z = z * (x - Cells(k, 1))
            Next k
            y = y + z
        Next j
        p = p + 1
        Cells(n + 2 + p, 1) = x
        Cells(n + 2 + p, 2) = y
    Next x
End Sub
```

```
Sub Auswertung()
    ReDim A(1, 1) As Double
    Dim n As Integer

    Call Werte_Lesen(n, A)
    Call Steigungen(n, A)
    Call Verlauf(n, A)
End Sub
```

```
Sub Verlauf_Zeigen()  
Range("A10:B60").Select  
Charts.Add  
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers  
ActiveChart.SetSourceData Source:=Sheets("Newton").Range("A10:B60"), PlotBy _  
:=xlColumns  
ActiveChart.Location Where:=xlLocationAsObject, Name:="Newton"  
With ActiveChart  
    .HasTitle = True  
    .ChartTitle.Characters.Text = "Seilverlauf"  
    .Axes(xlCategory, xlPrimary).HasTitle = True  
    .Axes(xlCategory, xlPrimary).AxisTitle.Characters.Text = "Weite [m]"  
    .Axes(xlValue, xlPrimary).HasTitle = True  
    .Axes(xlValue, xlPrimary).AxisTitle.Characters.Text = "Höhe [m]"  
End With  
ActiveChart.Legend.Select  
Selection.Delete  
End Sub
```

```
Sub Verlauf_Entfernen()  
Dim Shp As Shape  
For Each Shp In Worksheets("Newton").Shapes  
    Shp.Delete  
Next  
End Sub
```

```
Option Explicit
```

```
Sub Newton_Leer()
    Call Verlauf_Entfernen
    ThisWorkbook.Worksheets("Newton").Cells.Clear
End Sub
```

```
Sub Newton_Testdaten()
    Call Newton_Leer
    Cells(1, 1) = 0: Cells(1, 2) = 30
    Cells(2, 1) = 10: Cells(2, 2) = 18
    Cells(3, 1) = 20: Cells(3, 2) = 11.5
    Cells(4, 1) = 30: Cells(4, 2) = 10
    Cells(5, 1) = 35: Cells(5, 2) = 10.5
    Cells(6, 1) = 40: Cells(6, 2) = 12.5
    Cells(7, 1) = 50: Cells(7, 2) = 20
End Sub
```

```
Private Sub Werte_Lesen(n, A)
    Dim i, j As Integer
    ,
    'Bestimmung belegter Zeilen
    'und Definition der notwendigen Datenfelder
    Cells(Rows.Count, 1).End(xlUp).Select
    n = ActiveCell.Row
    ReDim A(n, n) As Double
    For i = 1 To n
        For j = 1 To 2
            A(i, j) = Cells(i, j)
        Next j
    Next i
End Sub
```

```
Private Sub Steigungen(n, A)
    Dim i, j As Integer

    For j = 3 To n + 1
        For i = 1 To n - j + 2
            Cells(i, j) = (Cells(i + 1, j - 1) - _
                Cells(i, j - 1)) / (Cells(i + j - 2, 1) - _
                Cells(i, 1))
        Next i
    Next j
    Exit Sub
End Sub
```

```
Private Sub Verlauf(n, A)
    Dim v, b, x, y, z As Double
    Dim i, j, k, p As Integer

    p = 0
    v = Cells(1, 1)
    b = Cells(n, 1)
    For x = v To b Step 1
        y = Cells(1, 2)
        For j = 3 To n + 1
            z = Cells(1, j)
            For k = 1 To j - 2
                z = z * (x - Cells(k, 1))
            Next k
            y = y + z
        Next j
        p = p + 1
        Cells(n + 2 + p, 1) = x
        Cells(n + 2 + p, 2) = y
    Next x
End Sub
```

```
Sub Auswertung()
    ReDim A(1, 1) As Double
    Dim n As Integer

    Call Werte_Lesen(n, A)
    Call Steigungen(n, A)
    Call Verlauf(n, A)
End Sub
```

```
Sub Verlauf_Zeigen()  
Range("A10:B60").Select  
Charts.Add  
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers  
ActiveChart.SetSourceData Source:= _  
    Sheets("Newton").Range("A10:B60"), PlotBy:= _  
    xlColumns  
ActiveChart.Location _  
    Where:=xlLocationAsObject, Name:="Newton"  
With ActiveChart  
    .HasTitle = True  
    .ChartTitle.Characters.Text = "Seilverlauf"  
    .Axes(xlCategory, xlPrimary).HasTitle = True  
    .Axes(xlCategory, _  
        xlPrimary).AxisTitle.Characters.Text = "Weite [m]"  
    .Axes(xlValue, xlPrimary).HasTitle = True  
    .Axes(xlValue, _  
        xlPrimary).AxisTitle.Characters.Text = "Höhe [m]"  
End With  
ActiveChart.Legend.Select  
Selection.Delete  
End Sub  
Sub Verlauf_Entfernen()  
Dim Shp As Shape  
For Each Shp In Worksheets("Newton").Shapes  
    Shp.Delete  
Next  
End Sub
```

```
Option Explicit
```

```
Sub LinAppro_Leer()
    ThisWorkbook.Worksheets _
        ("Lineare Approximation").Cells.Clear
End Sub
```

```
Sub LinAppro_Testdaten()
    Call LinAppro_Leer
    Cells(1, 1) = 20: Cells(1, 2) = 20
    Cells(2, 1) = 30: Cells(2, 2) = 55
    Cells(3, 1) = 40: Cells(3, 2) = 70
    Cells(4, 1) = 50: Cells(4, 2) = 75
    Cells(5, 1) = 65: Cells(5, 2) = 82
    Cells(6, 1) = 70: Cells(6, 2) = 86
    Cells(7, 1) = 80: Cells(7, 2) = 89
    Cells(8, 1) = 90: Cells(8, 2) = 90
    Cells(9, 1) = 100: Cells(9, 2) = 91
End Sub
```

```
Private Sub Werte_Lesen(n, A)
    Dim i, j As Integer
    ,
    'Bestimmung belegter Zeilen
    'und Definition der notwendigen Datenfelder
    Cells(Rows.Count, 1).End(xlUp).Select
    n = ActiveCell.Row
    ReDim A(n, 2) As Double
    For i = 1 To 2
        For j = 1 To 2
            A(i, j) = Cells(i, j)
        Next j
    Next i
End Sub
```

```
Sub LinAppro_Auswertung()
    ReDim A(1, 1) As Double
    Dim p, q, sx, sy, sxx, sxy, y As Double
    Dim n, i As Integer
    ,
    'Daten lesen
    Call Werte_Lesen(n, A)
    ,
    'Festpunktmethode
    p = (Cells(n, 2) - Cells(1, 2)) / _
        (Cells(n, 1) - Cells(1, 1))
    q = Cells(n, 2) - (Cells(n, 2) - Cells(1, 2)) / _
        (Cells(n, 1) - Cells(1, 1)) * Cells(n, 1)
    Cells(1, 7) = "y ="
    Cells(1, 8) = p
    Cells(1, 9) = "x +"
    Cells(1, 10) = q
    ,
    'Funktionswerte
    For i = 1 To n
        y = p * Cells(i, 1) + q
        Cells(i, 3) = y
    Next i
    ,
    'Kleinste Fehlerquadrate
    For i = 1 To n
        sx = sx + Cells(i, 1)
        sxx = sxx + Cells(i, 1) * Cells(i, 1)
        sy = sy + Cells(i, 2)
        sxy = sxy + Cells(i, 1) * Cells(i, 2)
    Next i
    p = (n * sxy - sx * sy) / (n * sxx - sx * sx)
    q = (sy * sxx - sxy * sx) / (n * sxx - sx * sx)
    Cells(3, 7) = "y ="
    Cells(3, 8) = p
    Cells(3, 9) = "x +"
    Cells(3, 10) = q
    ,
    'Funktionswerte
    For i = 1 To n
        y = p * Cells(i, 1) + q
        Cells(i, 4) = y
    Next i
End Sub
```



```
Next i
End Sub

Sub LinAppro_Diagramm()

Range("A1:D9").Select
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:=_
    Sheets("Lineare Approximation").Range( "A1:D9"), _
    PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = _
    "=""Sensorkennlinie""
ActiveChart.SeriesCollection(2).Name = _
    "=""Festpunktmethode""
ActiveChart.SeriesCollection(3).Name = _
    "=""Methode der kleinsten Fehlerquadrate""
ActiveChart.Location Where:=xlLocationAsObject, Name:= _
    "Lineare Approximation"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = _
        "Approximation einer Sensorkennlinie"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, _
        xlPrimary).AxisTitle.Characters.Text = "x/xmax %"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, _
        xlPrimary).AxisTitle.Characters.Text = "y/ymax %"
End With
ActiveWindow.Visible = False
End Sub

Sub LinAppro_Entfernen()
Dim Shp As Shape
For Each Shp In Worksheets("Lineare Approximation").Shapes
    Shp.Delete
Next
End Sub
```

Option Explicit

```

Sub Zugspannung_Leer()
  Dim Shp As Shape
  For Each Shp In Worksheets("Konstante Zugspannung").Shapes
    Shp.Delete
  Next
  ThisWorkbook.Worksheets("Konstante" & _
    "Zugspannung").Cells.Clear

  Range("A1:E1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Träger mit konstanter Zugspannung"
  Range("A2:A16").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("A2") = "A0 [m" & ChrW(178) & " ]"
  Range("A3") = "l [m]"
  Range("A4") = ChrW(961) & " [kg/m" & ChrW(179) & " ]"
  Range("A5") = ChrW(963) & " [N/m" & ChrW(178) & " ]"
  Range("A6") = "n"
  Range("B:B").ColumnWidth = "15"
  Range("C:C").ColumnWidth = "2"
  Range("D2") = "x [m]"
  Range("E2") = "A [m" & ChrW(178) & " ]"
  Range("D2:E2").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("B2").Select
End Sub

```

```

Sub Zugspannung_Testdaten()
  Cells(2, 2) = 0.2
  Cells(3, 2) = 0.5
  Cells(4, 2) = 0.00785
  Cells(5, 2) = 0.01
  Cells(6, 2) = 50
End Sub

```

```

Sub Zugspannung_Auswertung()
  Dim A0, l, r, S, n As Double
  Dim dx, x, dA, Ax As Double
  Dim i As Integer

  A0 = Cells(2, 2)
  l = Cells(3, 2)
  r = Cells(4, 2)
  S = Cells(5, 2)
  n = Cells(6, 2)
  dx = l / n
  Ax = A0
  i = 2
  For x = dx To l + dx Step dx
    dA = r * 9.81 / S * Ax * dx
    Ax = Ax + dA
    i = i + 1
    Cells(i, 4) = x
    Cells(i, 5) = Ax
  Next x
End Sub

```

```

Sub Zeige_Querschnittsverlauf()
  Range("D3:E52").Select
  Charts.Add
  ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
  ActiveChart.SetSourceData Source:= _
    Sheets("Konstante Zugspannung").Range("D3:E52"), _
    PlotBy:=xlColumns
  ActiveChart.SeriesCollection(1).Name = _
    "=""Querschnittsverlauf""
  ActiveChart.Location Where:= _
    xlLocationAsObject, Name:="Konstante Zugspannung"
  With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = _

```

```
    "Träger mit konstanter Spannung"  
.Axes(xlCategory, xlPrimary).HasTitle = True  
.Axes(xlCategory, _  
    xlPrimary).AxisTitle.Characters.Text = "x [mm]"  
.Axes(xlValue, xlPrimary).HasTitle = True  
.Axes(xlValue, _  
    xlPrimary).AxisTitle.Characters.Text = "A [mm^2]"  
End With  
ActiveChart.Legend.Select  
Selection.Left = 229  
Selection.Top = 274  
ActiveChart.Axes(xlValue).MajorGridlines.Select  
ActiveChart.PlotArea.Select  
Selection.Width = 314  
ActiveWindow.Visible = False  
End Sub  
  
Sub Lösche_Querschnittsverlauf()  
Dim Shp As Shape  
For Each Shp In Worksheets("Konstante Zugspannung").Shapes  
    Shp.Delete  
Next  
End Sub
```

Option Explicit

```

Sub Ausflusszeit_Leer()
  Dim Shp As Shape
  For Each Shp In Worksheets("Ausflusszeit").Shapes
    Shp.Delete
  Next
  ThisWorkbook.Worksheets("Ausflusszeit").Cells.Clear

  Range("A1:E1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = _
    "Ausflusszeit bei abnehmendem Flüssigkeitsstand"

  Range("A2:A7").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("A2") = "A [mm]" & ChrW(178) & " ]"
  Range("A3") = "h [mm]"
  Range("A4") = "u [mm]"
  Range("A5") = "n"
  Range("A7") = "t (TR) [s]"
  Range("B:B").ColumnWidth = "15"
  Range("C:C").ColumnWidth = "2"
  Range("D2") = "x [mm]"
  Range("E2") = "y1"
  Range("F2") = "y2"
  Range("G2") = "t1 [s]"
  Range("H2") = "t2 [s]"
  Range("I2") = "tm [s]"
  Range("J2") = "dtm [s]"
  Range("D2:J2").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True

  Range("B2").Select
End Sub

```

```

Sub Ausflusszeit_Testdaten()
  Cells(2, 2) = 10
  Cells(3, 2) = 400
  Cells(4, 2) = 10
  Cells(5, 2) = 50
End Sub

```

```

Sub Trichter_Form(h, x, Ax)
  Dim r As Double
  r = 100 + 100 / h * x
  Ax = r * r * 4 * Atn(1)
End Sub

```

```

Sub Ausflusszeit_Auswertung()
  Dim A, Ax, Ax1, Ax2, h, u, dx, x, y1, y2 As Double
  Dim k, dt, dt1, dt2, t, t1, t2, Su As Double
  Dim i, n As Integer

  A = Cells(2, 2)
  h = Cells(3, 2)
  u = Cells(4, 2)
  n = Cells(5, 2)
  i = 2

  dx = (h - u) / n
  t1 = 0: t2 = 0: t = 0
  k = 1 / (A * Sqr(2 * 9810))
  For x = h To u + dx Step -dx
    Call Trichter_Form(h, x, Ax1)
    Call Trichter_Form(h, x - dx, Ax2)
  Next x

  'Summation nach der Trapezregel
  y1 = Ax1 / Sqr(x)
  Su = Su + y1
  y2 = Ax2 / Sqr(x - dx)
  Su = Su + y2

```

```
'Bestimmung von dt aus dem Differenzenquotienten
```

```
dt1 = k * Ax1 / Sqr(x) * dx
t1 = t1 + dt1
dt2 = k * Ax2 / Sqr(x - dx) * dx
t2 = t2 + dt2
Ax = (Ax1 + Ax2) / 2
dt = k * Ax / Sqr(x - dx / 2) * dx
t = t + dt
```

```
'Ausgabe
```

```
i = i + 1
Cells(i, 4) = x
Cells(i, 5) = y1
Cells(i, 6) = y2
Cells(i, 7) = t1
Cells(i, 8) = t2
Cells(i, 9) = t
Cells(i, 10) = dt
```

```
Next x
```

```
'Bestimmung der Ausflusszeit nach der Trapezregel
```

```
t = Su * dx / 2 / (A * Sqr(2 * 9810))
```

```
Cells(7, 2) = t
```

```
End Sub
```

```
Sub Zeitdifferenzen_zeigen()
```

```
Range("D3:D51").Select
ActiveWindow.ScrollRow = 14
ActiveWindow.ScrollRow = 13
ActiveWindow.ScrollRow = 12
ActiveWindow.ScrollRow = 11
ActiveWindow.ScrollRow = 10
ActiveWindow.ScrollRow = 9
ActiveWindow.ScrollRow = 8
ActiveWindow.ScrollRow = 7
ActiveWindow.ScrollRow = 6
ActiveWindow.ScrollRow = 5
ActiveWindow.ScrollRow = 4
ActiveWindow.ScrollRow = 3
ActiveWindow.ScrollRow = 2
ActiveWindow.ScrollRow = 1
Range("D3:D51,J3:J51").Select
Range("J3").Activate
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData Source:= _
    Sheets("Ausflusszeit").Range( _
    "D3:D51,J3:J51"), PlotBy:=xlColumns
ActiveChart.Location Where:= _
    xlLocationAsObject, Name:="Ausflusszeit"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Zeitverhalten"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, _
    xlPrimary).AxisTitle.Characters.Text = "h [mm]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, _
    xlPrimary).AxisTitle.Characters.Text = "dt [s]"
End With
ActiveChart.HasLegend = False
```

```
End Sub
```

```
Sub Zeitdifferenzen_löschen()
```

```
Dim Shp As Shape
For Each Shp In Worksheets("Ausflusszeit").Shapes
    Shp.Delete
```

```
Next
```

```
End Sub
```

Option Explicit

```

Sub Schubkurbel_Leer()
  Dim Shp As Shape
  For Each Shp In Worksheets("Schubkurbeltrieb").Shapes
    Shp.Delete
  Next
  ThisWorkbook.Worksheets("Schubkurbeltrieb").Cells.Clear
  Range("A1:B1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Schubkurbeltrieb"

  Range("C:C").ColumnWidth = "2"
  Range("D1:E1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Indikatordiagramm"
  Range("D2") = ChrW(966) & " [Grad]"
  Range("E2") = "p [N/m" & ChrW(178) & "]"
  Range("D2:E2").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True

  Range("F:F").ColumnWidth = "2"
  Range("G1:H1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Auswertung"
  Range("G2") = "x [mm]"
  Range("H2") = "FK [N]"
  Range("I2") = ChrW(946) & " [Grad]"
  Range("J2") = "FST [N]"
  Range("K2") = "FR [N]"
  Range("L2") = ChrW(969) & " [1/s]"
  Range("M2") = ChrW(916) & "t [s]"
  Range("N2") = "t [s]"
  Range("O2") = "v [mm/s]"
  Range("P2") = "a [mm/s" & ChrW(178) & "]"
  Range("Q2") = "FO [N]"
  Range("R2") = "FT [N]"
  Range("S2") = "Md [Nmm]"
  Range("T2") = "W [Nm/s]"
  Range("G2:T2").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("G:T").Select
  Selection.NumberFormat = "0.00"

  Range("A2:A16").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("A2") = "d [mm]"
  Range("A3") = "l [mm]"
  Range("A4") = "r [mm]"
  Range("A5") = "rST [mm]"
  Range("A6") = "mST [kg]"
  Range("A7") = "mB [kg]"
  Range("A8") = "mZ [kg]"
  Range("A9") = "mK [kg]"
  Range("A10") = "rW [mm]"
  Range("A11") = "mW [kg]"
  Range("A12") = "mN [kg]"
  Range("A15") = "mO [kg]"
  Range("A16") = "mR [kg]"
  Range("B2:B20").Select
  Selection.NumberFormat = "0.00"
  Range("B2").Select
End Sub

Sub Schubkurbel_Testdaten()
  Cells(2, 2) = 100
  Cells(3, 2) = 300
  Cells(4, 2) = 50

```

```

Cells(5, 2) = 124
Cells(6, 2) = 60
Cells(7, 2) = 23.2
Cells(8, 2) = 6
Cells(9, 2) = 32
Cells(10, 2) = 12
Cells(11, 2) = 61.66
Cells(12, 2) = 4
Cells(3, 4) = -90: Cells(3, 5) = 100000
Cells(4, 4) = -80: Cells(4, 5) = 70000
Cells(5, 4) = -70: Cells(5, 5) = 45000
Cells(6, 4) = -60: Cells(6, 5) = 25000
Cells(7, 4) = -50: Cells(7, 5) = 20000
Cells(8, 4) = -40: Cells(8, 5) = 20000
Cells(9, 4) = -30: Cells(9, 5) = 20000
Cells(10, 4) = -20: Cells(10, 5) = 20000
Cells(11, 4) = -10: Cells(11, 5) = 20000
Cells(12, 4) = 0: Cells(12, 5) = 20000
Cells(13, 4) = 10: Cells(13, 5) = 20000
Cells(14, 4) = 20: Cells(14, 5) = 20000
Cells(15, 4) = 30: Cells(15, 5) = 20000
Cells(16, 4) = 40: Cells(16, 5) = 20000
Cells(17, 4) = 50: Cells(17, 5) = 20000
Cells(18, 4) = 60: Cells(18, 5) = 55000
Cells(19, 4) = 70: Cells(19, 5) = 100000
Cells(20, 4) = 80: Cells(20, 5) = 190000
Cells(21, 4) = 90: Cells(21, 5) = 600000
Cells(22, 4) = 100: Cells(22, 5) = 630000
Cells(23, 4) = 110: Cells(23, 5) = 700000
Cells(24, 4) = 120: Cells(24, 5) = 735000
Cells(25, 4) = 130: Cells(25, 5) = 750000
Cells(26, 4) = 140: Cells(26, 5) = 720000
Cells(27, 4) = 150: Cells(27, 5) = 650000
Cells(28, 4) = 160: Cells(28, 5) = 560000
Cells(29, 4) = 170: Cells(29, 5) = 480000
Cells(30, 4) = 180: Cells(30, 5) = 400000
Cells(31, 4) = 190: Cells(31, 5) = 350000
Cells(32, 4) = 200: Cells(32, 5) = 305000
Cells(33, 4) = 210: Cells(33, 5) = 280000
Cells(34, 4) = 220: Cells(34, 5) = 250000
Cells(35, 4) = 230: Cells(35, 5) = 230000
Cells(36, 4) = 240: Cells(36, 5) = 200000
Cells(37, 4) = 250: Cells(37, 5) = 180000
Cells(38, 4) = 260: Cells(38, 5) = 135000
Cells(39, 4) = 270: Cells(39, 5) = 100000

```

End Sub

Sub Schubkurbel_Auswertung()

```

Dim d, l, r, rST, mST, mB, mZ, mK, rW, mW, mN As Double
Dim mO, mR, ph, x, la, FK, p, be, FST, FR, z As Double
Dim w, dx, xa, dph, dt, t, v, va, dv, aK, FO As Double
Dim FT, Md, WA As Double
Dim i As Integer
d = Cells(2, 2)
l = Cells(3, 2)
r = Cells(4, 2)
rST = Cells(5, 2)
mST = Cells(6, 2)
mB = Cells(7, 2)
mZ = Cells(8, 2)
mK = Cells(9, 2)
rW = Cells(10, 2)
mW = Cells(11, 2)
mN = Cells(12, 2)

```

'Massenaufteilung

```

mO = mST * rST / l + mK + mB
mR = mST * (1 - rST) / l + mW * rW / r + mZ + mN
Cells(15, 2) = mO
Cells(16, 2) = mR

```

'Bewegung

'Konstante: Atn(1)=pi/4

```

la = r / l
xa = 0
dph = 10
t = 0
va = 0
WA = 0

```

```

For i = 3 To 39
  ph = Cells(i, 4)
  p = Cells(i, 5)
  ph = ph / 45 * Atn(1)
  x = r * (1 - Sin(ph)) + _
    l * (1 - Sqr(1 - (la * Cos(ph)) ^ 2))
  If i = 3 Then xa = x
  dx = x - xa
  xa = x
  Cells(i, 7) = x
  FK = d ^ 2 * Atn(1) * p / 1000000
  Cells(i, 8) = FK
  be = r * Cos(ph) / l
  If 1 - be * be >= 0 Then
    be = Atn(be / Sqr(1 - be * be))
  Else
    be = 0
  End If
  Cells(i, 9) = be / Atn(1) * 45
  FST = FK / Cos(be)
  Cells(i, 10) = FST
  FR = FST * Cos(2 * Atn(1) - ph - be)
  Cells(i, 11) = FR
  w = Sqr(Abs(FR / (mR * r)) * 9810)
  Cells(i, 12) = w
  If Not w = 0 Then
    dt = Abs(dph / w)
  Else
    dt = 0
  End If
  Cells(i, 13) = dt
  t = t + dt
  Cells(i, 14) = t
  If dt > 0 Then
    v = dx / dt
  Else
    v = 0
  End If
  Cells(i, 15) = v
  If i = 3 Then va = v
  dv = v - va
  va = v
  If dt > 0 Then
    aK = dv / dt
  Else
    aK = 0
  End If
  Cells(i, 16) = aK
  FO = m0 * aK / 9810
  Cells(i, 17) = FO
  FT = FST * Sin(2 * Atn(1) - ph - be)
  Cells(i, 18) = FT
  Md = FT * r
  Cells(i, 19) = Md
  WA = WA + Md * dph / 1000
  Cells(i, 20) = WA
Next i
End Sub

```


Option Explicit

```

Sub Drehschwingung_Leer()
  Dim Shp As Shape
  For Each Shp In Worksheets("Drehschwingung").Shapes
    Shp.Delete
  Next
  ThisWorkbook.Worksheets("Drehschwingung").Cells.Clear

  Range("A1:B1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Drehschwingung"

  Range("A2:A16").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Range("A2") = "G [N/m" & ChrW(178) & "]"
  Range("A3") = "Ip [m^4]"
  Range("A4") = "l [m]"
  Range("A5") = "Id [kgm]"
  Range("A6") = ChrW(966) & "[kg]"
  Range("A7") = ChrW(969) & "[1/s]"
  Range("A8") = "t0 [s]"
  Range("A9") = ChrW(916) & "t [s]"
  Range("A10") = "tmax [s]"
  Range("B:B").ColumnWidth = "15"

  Range("C:C").ColumnWidth = "2"
  Range("D1:F1").Select
  Selection.MergeCells = True
  Selection.Font.Bold = True
  Selection.Font.Italic = True
  Selection.Value = "Auswertung"
  Range("D2") = "t [s]"
  Range("E2") = ChrW(969) & "[1/s]"
  Range("F2") = ChrW(966) & "[Grad]"
  Range("D2:F2").Select
  Selection.Font.Bold = True
  Selection.Font.Italic = True

  Range("B2").Select
End Sub

Sub Drehschwingung_Testdaten()
  Cells(2, 2) = 80000000000#
  Cells(3, 2) = 0.0000002
  Cells(4, 2) = 0.6
  Cells(5, 2) = 4.5
  Cells(6, 2) = 4
  Cells(7, 2) = 0
  Cells(8, 2) = 0
  Cells(9, 2) = 0.005
  Cells(10, 2) = 0.2
End Sub

Sub Drehschwingung_Auswertung()
  Dim G, Ip, l, Id, p0, w0, t0, dt, tm As Double
  Dim t, p, dp, dw, w As Double
  Dim i As Integer

  G = Cells(2, 2)
  Ip = Cells(3, 2)
  l = Cells(4, 2)
  Id = Cells(5, 2)
  p0 = Cells(6, 2)
  w0 = Cells(7, 2)
  t0 = Cells(8, 2)
  dt = Cells(9, 2)
  tm = Cells(10, 2)
  t = t0
  p = p0
  w = w0
  i = 2
  Do
    i = i + 1

```

```
t = t + dt
Cells(i, 4) = t
dw = -G * Ip / l / Id * p * dt
w = w + dw
Cells(i, 5) = w
dp = w * dt
p = p + dp
Cells(i, 6) = p
Loop While t < tm
End Sub

Sub Drehschwingung_Zeigen()
Range("D3:D42,F3:F42").Select
Range("F3").Activate
Charts.Add
ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
ActiveChart.SetSourceData _
    Source:=Sheets("Drehschwingung").Range( _
    "D3:D42,F3:F42"), PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = _
    "=""Auslenkung""
ActiveChart.Location Where:=xlLocationAsObject, _
    Name:="Drehschwingung"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Drehschwingung"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, _
        xlPrimary).AxisTitle.Characters.Text = "t [s]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, xlPrimary).AxisTitle.Characters. _
        Text = "Winkel [Grad]"
End With
ActiveWindow.Visible = False
Windows("Kapitel 05.xls").Activate
End Sub

Sub Drehschwingung_Löschen()
Dim Shp As Shape
For Each Shp In Worksheets("Drehschwingung").Shapes
    Shp.Delete
Next
End Sub
```

Option Explicit

```
Sub Membran_Formular()
  Dim Shp As Shape
  For Each Shp In Worksheets("Membran").Shapes
    Shp.Delete
  Next
  ThisWorkbook.Worksheets("Membran").Cells.Clear
```

```
Cells(1, 6) = 0
Cells(2, 5) = 0
Cells(2, 7) = 0
Cells(3, 4) = 0
Cells(3, 8) = 0
Cells(4, 3) = 0
Cells(4, 9) = 0
Cells(5, 2) = 0
Cells(5, 10) = 0
Cells(6, 1) = 0
Cells(6, 11) = 0
Cells(7, 2) = 0
Cells(7, 10) = 0
Cells(8, 3) = 0
Cells(8, 9) = 0
Cells(9, 4) = 0
Cells(9, 8) = 0
Cells(10, 5) = 0
Cells(10, 7) = 0
Cells(11, 6) = 0
```

```
Cells(4, 6) = 1
Cells(5, 5) = 1
Cells(5, 7) = 1
Cells(6, 4) = 1
Cells(6, 8) = 1
Cells(7, 5) = 1
Cells(7, 7) = 1
Cells(8, 6) = 1
```

```
Range("I6:J6").Select
With Selection.Interior
  .ColorIndex = 15
  .Pattern = xlSolid
End With
Range("H5:I5").Select
With Selection.Interior
  .ColorIndex = 15
  .Pattern = xlSolid
End With
Range("H4").Select
With Selection.Interior
  .ColorIndex = 15
  .Pattern = xlSolid
End With
Range("A12").Select
With Selection.Interior
  .ColorIndex = 15
  .Pattern = xlSolid
End With
```

End Sub

```
Sub Membran_Auswertung()
  Dim u40, u30, u31, u21, u22 As Double
  Dim v40, v30, v31, v21, v22 As Double
  Dim w40, w30, w31, w21, w22 As Double
  Dim i, n As Integer
  n = Cells(12, 1)
  u40 = Cells(6, 10)
  u30 = Cells(6, 9)
  u31 = Cells(5, 9)
  u21 = Cells(5, 8)
  u22 = Cells(4, 8)
```

```
For i = 1 To n
  v40 = 1 / 15 * (2 * u31 + 1)
  v30 = 4 * v40
  v31 = 1 / 15 * (u22 + 4 * u30 + 2)
```

```
v21 = 4 * v31 - u30
v22 = 1 / 4 * (v21 + u21)
```

```
w22 = 1 / 15 * (4 * v21 + v31 + 2)
w21 = 4 * w22 - v21
w31 = 1 / 14 * (4 * v21 + v40 + 1)
w30 = 4 * w31 - v21
w40 = 1 / 4 * v30
```

```
u40 = w40
u30 = w30
u31 = w31
u21 = w21
u22 = w22
```

```
Cells(6, 10) = u40
Cells(6, 9) = u30
Cells(5, 9) = u31
Cells(5, 8) = u21
Cells(4, 8) = u22
```

```
Next i
```

```
Übertragung durch Symmetrie
```

```
Cells(7, 9) = Cells(5, 9)
Cells(7, 8) = Cells(5, 8)
Cells(8, 8) = Cells(4, 8)
Cells(8, 7) = Cells(4, 7)
Cells(9, 7) = Cells(3, 7)
Cells(9, 6) = Cells(3, 6)
Cells(10, 6) = Cells(2, 6)
Cells(2, 6) = Cells(6, 10)
Cells(3, 6) = Cells(6, 9)
Cells(3, 7) = Cells(5, 9)
Cells(4, 7) = Cells(5, 8)
Cells(3, 5) = Cells(3, 7)
Cells(4, 5) = Cells(4, 7)
Cells(8, 5) = Cells(8, 7)
Cells(9, 5) = Cells(9, 7)
Cells(4, 4) = Cells(4, 8)
Cells(5, 4) = Cells(5, 8)
Cells(7, 4) = Cells(7, 8)
Cells(8, 4) = Cells(8, 8)
Cells(5, 3) = Cells(5, 9)
Cells(6, 3) = Cells(6, 9)
Cells(7, 3) = Cells(7, 9)
Cells(6, 2) = Cells(6, 10)
```

```
End Sub
```

Option Explicit

```
'Prozedur Matrix A neu
Sub Matrix_A_Neu()
    Dim Blatt As Worksheet
    Dim Name As String

    Name = "Matrix A"
    On Error GoTo Matrix_A_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear
    Exit Sub
Matrix_A_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub
```

```
'Prozedur Matrix B neu
Sub Matrix_B_Neu()
    Dim Blatt As Worksheet
    Dim Name As String

    Name = "Matrix B"
    On Error GoTo Matrix_B_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear
    Exit Sub
Matrix_B_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub
```

```
'Prozedur Matrix C neu
Sub Matrix_C_Neu()
    Dim Blatt As Worksheet
    Dim Name As String

    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear
    Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub
```

```
Sub Matrix_Transponierte()
    Dim Blatt As Worksheet
    Dim Name As String
    Dim i, k, m, Zeilen, Spalten As Integer
    Dim A() As Double

    Name = "Matrix A"
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Zeilen = Blatt.UsedRange.Rows.Count
    Spalten = Blatt.UsedRange.Columns.Count
```

```
'Matrix A lesen
ReDim A(Zeilen, Spalten)
For i = 1 To Zeilen
    For k = 1 To Spalten
        A(i, k) = Blatt.Cells(i, k)
    Next k
Next i
```

```
'Überprüfung, ob Matrix B vorhanden
Name = "Matrix B"
m = 0
For Each Blatt In Sheets
    If Name = Blatt.Name Then
```

```
        m = 1
    End If
Next

'Matrix B anlegen
If m = 0 Then
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
End If
Set Blatt = ThisWorkbook.Worksheets(Name)
Blatt.Activate
Blatt.Cells.Clear

'Bestimmung der Transponierten
For i = 1 To Zeilen
    For k = 1 To Spalten
        Blatt.Cells(k, i) = A(i, k)
    Next k
Next i
End Sub
```

```

Sub Matrix_Addition()
    Dim Blatt1, Blatt2, Blatt As Worksheet
    Dim Name1, Name2, Name As String
    Dim i, k, m, Z1, Z2, S1, S2 As Integer
    Dim A(), B() As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung, ob Matrix B vorhanden
    Name2 = "Matrix B"
    Set Blatt2 = ThisWorkbook.Worksheets(Name2)
    m = 0
    For Each Blatt In Sheets
        If Name2 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix B fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z2 = Blatt2.UsedRange.Rows.Count
    S2 = Blatt2.UsedRange.Columns.Count

'Matrix B lesen
    ReDim B(Z2, S2)
    For i = 1 To Z2
        For k = 1 To S2
            B(i, k) = Blatt2.Cells(i, k)
        Next k
    Next i

'Überprüfung der Typen
    If Z1 <> Z2 Or S1 <> S2 Then
        MsgBox "Matrixtypen unterschiedlich!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If

'Überprüfung, ob Matrix C vorhanden
    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Addition
    For i = 1 To Z1
        For k = 1 To S1
            Blatt.Cells(i, k) = A(i, k) + B(i, k)
        Next k
    Next i

    Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add

```

Blatt.Name = Name

Resume

End Sub


```

Sub Matrix_Subtraktion()
    Dim Blatt1, Blatt2, Blatt As Worksheet
    Dim Name1, Name2, Name As String
    Dim i, k, m, Z1, Z2, S1, S2 As Integer
    Dim A(), B() As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung, ob Matrix B vorhanden
    Name2 = "Matrix B"
    Set Blatt2 = ThisWorkbook.Worksheets(Name2)
    m = 0
    For Each Blatt In Sheets
        If Name2 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix B fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z2 = Blatt2.UsedRange.Rows.Count
    S2 = Blatt2.UsedRange.Columns.Count

'Matrix B lesen
    ReDim B(Z2, S2)
    For i = 1 To Z2
        For k = 1 To S2
            B(i, k) = Blatt2.Cells(i, k)
        Next k
    Next i

'Überprüfung der Typen
    If Z1 <> Z2 Or S1 <> S2 Then
        MsgBox "Matrixtypen unterschiedlich!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If

'Überprüfung, ob Matrix C vorhanden
    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Addition
    For i = 1 To Z1
        For k = 1 To S1
            Blatt.Cells(i, k) = A(i, k) - B(i, k)
        Next k
    Next i
    Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub

```



```

Sub Matrix_Skalarprodukt()
    Dim Blatt1, Blatt2, Blatt As Worksheet
    Dim Name1, Name2, Name As String
    Dim i, k, m, Z1, S1 As Integer
    Dim A() As Double
    Dim Skalar As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung, ob Matrix B vorhanden
    Name2 = "Matrix B"
    Set Blatt2 = ThisWorkbook.Worksheets(Name2)
    m = 0
    For Each Blatt In Sheets
        If Name2 = Blatt.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Skalar in Matrix B fehlt!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If
    Skalar = Blatt2.Cells(1, 1)

'Überprüfung, ob Matrix C vorhanden
    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Addition
    For i = 1 To Z1
        For k = 1 To S1
            Blatt.Cells(i, k) = A(i, k) * Skalar
        Next k
    Next i
    Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub

```

```

Sub Matrix_Produkt()
    Dim Blatt1, Blatt2, Blatt As Worksheet
    Dim Name1, Name2, Name As String
    Dim i, j, k, m, Z1, Z2, S1, S2 As Integer
    Dim A(), B() As Double
    Dim Sum As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt1.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung, ob Matrix B vorhanden
    Name2 = "Matrix B"
    Set Blatt2 = ThisWorkbook.Worksheets(Name2)
    m = 0
    For Each Blatt In Sheets
        If Name2 = Blatt2.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix B fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z2 = Blatt2.UsedRange.Rows.Count
    S2 = Blatt2.UsedRange.Columns.Count

'Matrix B lesen
    ReDim B(Z2, S2)
    For i = 1 To Z2
        For k = 1 To S2
            B(i, k) = Blatt2.Cells(i, k)
        Next k
    Next i

'Überprüfung ob Multiplikation zulässig
    If S1 <> Z2 Then
        MsgBox "Multiplikation nicht möglich!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If

'Überprüfung, ob Matrix C vorhanden
    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Produktbildung
    For j = 1 To S2
        For i = 1 To Z1
            Sum = 0
            For k = 1 To S1
                Sum = Sum + A(i, k) * B(k, j)
            Next k
            Blatt.Cells(i, j) = Str(Sum)
        Next i
    Next j

```

```
        Next i
    Next j
Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub
```

```

Sub Matrix_Determinante()
    Dim Blatt1, Blatt As Worksheet
    Dim Name1, Name As String
    Dim i, j, k, m, n, Z1, S1 As Integer
    Dim A() As Double
    Dim Sum, Pro, q As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt1.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung ob quadratische Matrix vorliegt
    If Z1 <> S1 Then
        MsgBox "Keine quadratische Matrix!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If

'Überprüfung, ob Matrix C vorhanden
    Name = "Matrix C"
    On Error GoTo Matrix_C_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Determinante
    q = Z1 / 2
    If q = Int(q) Then
        n = Z1 - 1
    Else
        n = Z1
    End If
    Sum = 0
    For i = 1 To n
        j = i
        Pro = 1
        For k = 1 To Z1
            Pro = Pro * A(k, j)
            j = j + 1
            If j > Z1 Then j = j - Z1
        Next k
        Sum = Sum + Pro
    Next i
    For i = 1 To n
        j = Z1 - i + 1
        Pro = 1
        For k = 1 To Z1
            Pro = Pro * A(k, j)
            j = j - 1
            If j < 1 Then j = j + Z1
        Next k
        Sum = Sum - Pro
    Next i
    Blatt.Cells(1, 1) = Sum
    Exit Sub
Matrix_C_Neu:
    Set Blatt = Worksheets.Add

```

Blatt.Name = Name

Resume

End Sub

```

Sub Matrix_Komplement()
    Dim Blatt1, Blatt As Worksheet
    Dim Name1, Name As String
    Dim i, j, k, m, n, Z1, S1 As Integer
    Dim j1, l, l1 As Integer
    Dim A(), B() As Double
    Dim Det, q As Double

'Überprüfung, ob Matrix A vorhanden
    Name1 = "Matrix A"
    Set Blatt1 = ThisWorkbook.Worksheets(Name1)
    m = 0
    For Each Blatt In Sheets
        If Name1 = Blatt1.Name Then
            m = 1
        End If
    Next
    If m = 0 Then
        MsgBox "Matrix A fehlt!", vbOKOnly, "ACHTUNG"
        Exit Sub
    End If
    Z1 = Blatt1.UsedRange.Rows.Count
    S1 = Blatt1.UsedRange.Columns.Count

'Matrix A lesen
    ReDim A(Z1, S1)
    For i = 1 To Z1
        For k = 1 To S1
            A(i, k) = Blatt1.Cells(i, k)
        Next k
    Next i

'Überprüfung, ob quadratische Matrix vorliegt
    If Z1 <> S1 Then
        MsgBox "Keine quadratische Matrix!", vbOKOnly, _
            "ACHTUNG"
        Exit Sub
    End If

'Überprüfung, ob Matrix B vorhanden
    Name = "Matrix B"
    On Error GoTo Matrix_B_Neu
    Set Blatt = ThisWorkbook.Worksheets(Name)
    Blatt.Activate
    Blatt.Cells.Clear

'Determinante
    q = Z1 / 2
    If q = Int(q) Then
        n = Z1 - 1
    Else
        n = Z1
    End If
    For i = 1 To n
        For k = 1 To n
            'Bildung der Untermatrix
            ReDim B(n - 1, n - 1)
            For j = 1 To n
                For l = 1 To n
                    If j <> i And l <> k Then
                        j1 = j
                        If j1 > i Then j1 = j1 - 1
                        l1 = l
                        If l1 > k Then l1 = l1 - 1
                        B(j1, l1) = A(j, l)
                    End If
                Next l
            Next j
            Call Matrix_Kofaktoren(B, n - 1, Det)
            Blatt.Cells(i, k) = (-1) ^ (i + k) * Det
        Next k
    Next i
    Exit Sub
Matrix_B_Neu:
    Set Blatt = Worksheets.Add
    Blatt.Name = Name
    Resume
End Sub

```



```
Sub Matrix_Kofaktoren(A, n, Det)
  Dim i, j, k, r As Integer
  Dim Pro, q As Double

  Det = 0
  q = n / 2
  If q = Int(q) Then
    r = n - 1
  Else
    r = n
  End If
  For i = 1 To r
    j = i
    Pro = 1
    For k = 1 To n
      Pro = Pro * A(k, j)
      j = j + 1
      If j > n Then j = j - n
    Next k
    Det = Det + Pro
  Next i

  For i = 1 To r
    j = n - i + 1
    Pro = 1
    For k = 1 To n
      Pro = Pro * A(k, j)
      j = j - 1
      If j < 1 Then j = j + n
    Next k
    Det = Det - Pro
  Next i
End Sub
```

```
Option Explicit
Dim A(), y(), x() As Double
Dim m, n As Integer
```

```
Sub Gauss_Leer()
    ThisWorkbook.Worksheets("Gauss").Cells.Clear
    Range("A1").Select
End Sub
```

```
Sub Gauss_Testdaten()
    Dim i, j As Integer
    For i = 1 To 15
        For j = 1 To 15
            Cells(i, j) = 0
            If i = j Then Cells(i, j) = 4
        Next j
    Next i
    Cells(1, 2) = -1: Cells(1, 6) = -1
    Cells(2, 1) = -1: Cells(2, 3) = -1: Cells(2, 7) = -1
    Cells(3, 2) = -1: Cells(3, 4) = -1: Cells(3, 8) = -1
    Cells(4, 3) = -1: Cells(4, 5) = -1: Cells(4, 9) = -1
    Cells(5, 4) = -1: Cells(5, 10) = -1
    Cells(6, 1) = -1: Cells(6, 7) = -1: Cells(6, 11) = -1
    Cells(7, 2) = -1: Cells(7, 6) = -1
    Cells(7, 8) = -1: Cells(7, 12) = -1
    Cells(8, 3) = -1: Cells(8, 7) = -1
    Cells(8, 9) = -1: Cells(8, 13) = -1
    Cells(9, 4) = -1: Cells(9, 8) = -1
    Cells(9, 10) = -1: Cells(9, 14) = -1
    Cells(10, 5) = -1: Cells(10, 9) = -1
    Cells(10, 15) = -1
    Cells(11, 6) = -1: Cells(11, 12) = -1
    Cells(12, 7) = -1: Cells(12, 11) = -1
    Cells(12, 13) = -1
    Cells(13, 8) = -1: Cells(13, 12) = -1
    Cells(13, 14) = -1
    Cells(14, 9) = -1: Cells(14, 13) = -1
    Cells(14, 15) = -1
    Cells(15, 10) = -1: Cells(15, 14) = -1
    Cells(1, 17) = 24
    Cells(2, 17) = 18
    Cells(3, 17) = 24
    Cells(4, 17) = 36
    Cells(5, 17) = 84
    Cells(6, 17) = 22
    Cells(7, 17) = 0
    Cells(8, 17) = 0
    Cells(9, 17) = 0
    Cells(10, 17) = 30
    Cells(11, 17) = 34
    Cells(12, 17) = 24
    Cells(13, 17) = 32
    Cells(14, 17) = 24
    Cells(15, 17) = 32
End Sub
```

```
Sub Gauss_Daten_lesen()
    Dim MyDoc As Object
    Dim nRows, nCols, i, j As Integer

    Set MyDoc = ThisWorkbook.Worksheets("Gauss")
    nRows = MyDoc.UsedRange.Rows.Count
    nCols = MyDoc.UsedRange.Columns.Count
    m = nRows
    n = nCols - 2
    ReDim A(m, n), y(m), x(n)

    For i = 1 To m
        y(i) = Cells(i, nCols)
    Next i
    For i = 1 To m
        For j = 1 To n
            A(i, j) = Cells(i, j)
        Next j
    Next i
End Sub
```

```

Sub Gauss_Elimination()
  Dim c As Double
  Dim i, j, k As Integer

  For i = 1 To m - 1
    For j = i + 1 To m
      c = A(j, i) / A(i, i)
      For k = 1 To n
        A(j, k) = A(j, k) - c * A(i, k)
      Next k
      y(j) = y(j) - c * y(i)
    Next j
  Next i
End Sub

```

```

Sub Gauss_Rückwärtsrechnung()
  Dim i, j, k As Integer
  Dim s As Double

  For i = n To 1 Step -1
    s = y(i)
    For k = i + 1 To n
      s = s - A(i, k) * x(k)
    Next k
    x(i) = s / A(i, i)
  Next i
End Sub

```

```

Sub Gauss_Auswertung()
  Dim i, j As Integer
  Dim s As Double

  Call Gauss_Daten_lesen

  Call Gauss_Elimination

  Call Gauss_Rückwärtsrechnung

```

```

'Ausgabe
For i = 1 To m
  For j = 1 To n
    Cells(m + 1 + i, j) = A(i, j)
  Next j
  Cells(m + 1 + i, n + 2) = y(i)
Next i
For j = 1 To n
  Cells(2 * m + 3, j) = x(j)
Next j

```

```

'Testrechnung
s = 0
For j = 1 To n
  s = s + A(1, j) * x(j)
Next j
Cells(2 * m + 3, n + 2) = s
End Sub

```

```

Option Explicit
Dim a(), x() As Double
Dim n As Integer
Sub vMises_Leer()
    ThisWorkbook.Worksheets("vMises").Cells.Clear
    Range("A1").Select
End Sub

Sub vMises_Testdaten()
    Cells(1, 1) = 7
    Cells(1, 2) = -4
    Cells(1, 3) = 1
    Cells(1, 4) = 0
    Cells(2, 1) = -4
    Cells(2, 2) = 6
    Cells(2, 3) = -4
    Cells(2, 4) = 1
    Cells(3, 1) = 1
    Cells(3, 2) = -4
    Cells(3, 3) = 6
    Cells(3, 4) = -3
    Cells(4, 1) = 0
    Cells(4, 2) = 1
    Cells(4, 3) = -3
    Cells(4, 4) = 2

    Cells(1, 6) = 1
    Cells(2, 6) = 3
    Cells(3, 6) = 3
    Cells(4, 6) = 4
End Sub

Sub vMises_Daten_lesen()
    Dim MyDoc As Object
    Dim nRows, nCols, i, j As Integer

    Set MyDoc = ThisWorkbook.Worksheets("vMises")
    nRows = MyDoc.UsedRange.Rows.Count
    n = nRows
    ReDim a(n, n), x(n)

    For i = 1 To n
        x(i) = Cells(i, n + 2)
    Next i
    For i = 1 To n
        For j = 1 To n
            a(i, j) = Cells(i, j)
        Next j
    Next i
End Sub

Sub vMises_Auswertung()
    Dim y(), z() As Double
    Dim i, j, k, m As Integer
    Dim L, p As Double

    Call vMises_Daten_lesen
    ReDim y(n), z(n)
    m = 10

'Iterationen
    For i = 1 To m
        For j = 1 To n
            y(j) = 0
            For k = 1 To n
                y(j) = y(j) + a(j, k) * x(k)
            Next k
        Next j

        'Quotient
        For j = 1 To n
            If x(j) = 0 Then
                Cells(i, n + 5 + j) = ""
            Else
                Cells(i, n + 5 + j) = y(j) / x(j)
            End If
        Next j
    
```

```
For j = 1 To n
    x(j) = y(j)
    Cells(j, n + 2) = y(j)
Next j
Next i
```

'Mittelwert bilden

```
L = 0
For i = 1 To n
    L = L + Cells(m, n + 5 + i)
Next i
L = L / n
Cells(m + 1, n + 5 + n) = L
```

'Eigenvektor

```
For i = 1 To n
    y(i) = 1 / L ^ m * x(i)
    Cells(i, n + 3) = y(i)
Next i
```

'Überprüfung

```
For i = 1 To n
    z(i) = 0
    For j = 1 To n
        p = a(i, j)
        If i = j Then p = p - L
        z(i) = z(i) + p * y(j)
    Next j
Next i
For i = 1 To n
    Cells(i, n + 4) = z(i)
Next i
```

End Sub

Option Explicit

```
Sub MonteCarlo_Test()  
  Dim x, y, z As Double  
  Dim n, m As Long  
  Dim i, j As Long  
  
  ThisWorkbook.Worksheets("MonteCarlo").Cells.Clear  
  n = 1000000  
  Randomize  
  Cells(1, 2) = Atn(1)  
  z = Timer  
  
  For i = 1 To 20  
    m = 0  
    For j = 1 To n  
      x = Rnd(z)  
      y = Rnd(z)  
      If x * x + y * y < 1 Then  
        m = m + 1  
      End If  
    Next j  
    Cells(i, 1) = Str(m / n)  
  Next i  
End Sub
```

Option Explicit

```
Sub MonteCarlo_Blechteil()  
  Dim x, y, z, e, A As Double  
  Dim n, m As Long  
  Dim i, j As Long  
  
  ThisWorkbook.Worksheets("Blechteil").Cells.Clear  
  n = InputBox("Anzahl Durchläufe:")  
  Randomize  
  A = 160 * 160  
  z = Timer  
  e = 1E-308  
  Cells(1, 1) = A * Atn(1)  
  For i = 1 To 20  
    m = 0  
    For j = 1 To n  
      x = (160 + e) * Rnd(z) - 80  
      y = (160 + e) * Rnd(z) - 80  
      If Sqr(x * x + y * y) <= 80 Then  
        If y <= 65 Then  
          m = m + 1  
        End If  
      End If  
    Next j  
    Cells(i, 2) = Str(A * m / n)  
  Next i  
End Sub
```

Option Explicit

```
Sub Maschinenwartung_Leer()
    Dim Shp As Shape
    For Each Shp In Worksheets("Maschinenwartung").Shapes
        Shp.Delete
    Next
    ThisWorkbook.Worksheets("Maschinenwartung").Cells.Clear
    Range("A1:B1").Select
    Selection.MergeCells = True
    Selection.Font.Bold = True
    Selection.Font.Italic = True
    Selection.Value = "Maschinenwartung"
```

```
Range("A:B").ColumnWidth = "10"
Range("C:C").ColumnWidth = "2"
Range("A2:A16").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Range("A2") = ChrW(916) & "t [s]"
Range("A3") = "tmax [s]"
Range("A4") = "x"
Range("A5") = "y"
```

```
Range("D1:E1").Select
Selection.MergeCells = True
Selection.Font.Bold = True
Selection.Font.Italic = True
Selection.Value = "Auswertung"
Range("D2:E2").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Range("D2") = "t [s]"
Range("E2") = "w"
Selection.NumberFormat = "0"
Range("B2").Select
```

End Sub

```
Sub Maschinenwartung_Testdaten()
    Cells(2, 2) = 1
    Cells(3, 2) = 360
    Cells(4, 2) = 0.3
    Cells(5, 2) = 0.4
```

End Sub

```
Sub Maschinenwartung_Auswertung()
    Dim w, t, dt, tm, i As Long
    Dim x, y, z As Double
```

```
dt = Cells(2, 2)
tm = Cells(3, 2)
x = Cells(4, 2)
y = Cells(5, 2)
```

Randomize

z = Timer

i = 2

w = 0

For t = 0 To tm Step dt

z = Rnd(z)

If z <= x Then

w = w + 1

End If

If w > 0 Then

z = Rnd(z)

If z <= y Then

w = w - 1

End If

End If

i = i + 1

Cells(i, 4) = t

Cells(i, 5) = w

Next t

End Sub

```
Sub Warteschlange_zeigen()
    Range("D3:E363").Select
```



```
Charts.Add
ActiveChart.ChartType = xlXYScatterLinesNoMarkers
ActiveChart.SetSourceData Source:= _
    Sheets("Maschinenwartung").Range("D3:E363"), _
    PlotBy:=xlColumns
ActiveChart.SeriesCollection(1).Name = _
    "=""Warteschlange""
ActiveChart.Location Where:= _
    xlLocationAsObject, Name:="Maschinenwartung"
With ActiveChart
    .HasTitle = True
    .ChartTitle.Characters.Text = "Warteschlange"
    .Axes(xlCategory, xlPrimary).HasTitle = True
    .Axes(xlCategory, _
        xlPrimary).AxisTitle.Characters.Text = _
        "t [Sek.]"
    .Axes(xlValue, xlPrimary).HasTitle = True
    .Axes(xlValue, _
        xlPrimary).AxisTitle.Characters.Text = _
        "Anzahl Maschinen"
End With
ActiveWindow.Visible = False
End Sub

Sub Grafik_löschen()
    Dim Shp As Shape
    For Each Shp In Worksheets("Maschinenwartung").Shapes
        Shp.Delete
    Next
End Sub
```

Option Explicit

```
Sub Ersatzproblem_Leer()
    Dim Shp As Shape
    For Each Shp In Worksheets("Ersatzproblem").Shapes
        Shp.Delete
    Next
    ThisWorkbook.Worksheets("Ersatzproblem").Cells.Clear
```

```
Range("A1:B1").Select
Selection.MergeCells = True
Selection.Font.Bold = True
Selection.Font.Italic = True
Selection.Value = "Instandhaltungsdaten"
```

```
Range("A:A").ColumnWidth = "30"
Range("B:B").ColumnWidth = "10"
Range("C:C").ColumnWidth = "2"
Range("D:I").ColumnWidth = "10"
Range("A2:A20").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Range("A2") = ChrW(916) & "t [100h]"
Range("A3") = "tmax [100h]"
Range("A4") = "tNutz [100h]"
Range("A5") = "Pumpe ausschalten [h]"
Range("A6") = "Anschlüsse demontieren [h]"
Range("A7") = "Gruppe demontieren [h]"
Range("A8") = "Ventil demontieren [h]"
Range("A9") = "Ventil reparieren [h]"
Range("A10") = "Ventil montieren [h]"
Range("A11") = "Gruppe montieren [h]"
Range("A12") = "Anschlüsse montieren [h]"
Range("A13") = "Pumpe einschalten [h]"
Range("A15") = "Auswertung"
Range("A16") = "Montagezeit [h]"
Range("A17") = "Ausfallzeit [h]"
Range("A18") = "Standzeit Einlass [100h]"
Range("A19") = "Standzeit Auslass [100h]"
Range("B2:B20").Select
Selection.NumberFormat = "0.0"
```

```
Range("D1:I1").Select
Selection.MergeCells = True
Selection.Font.Bold = True
Selection.Font.Italic = True
Selection.Value = "Auswertung"
Range("D2:j2").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Selection.NumberFormat = "0"
Range("D2") = "E1 [100h]"
Range("E2") = "E2 [100h]"
Range("F2") = "E3 [100h]"
Range("G2") = "A1 [100h]"
Range("H2") = "A2 [100h]"
Range("I2") = "A3 [100h]"
Range("B2").Select
```

End Sub

```
Sub Ersatzproblem_Testdaten()
    Cells(2, 2) = 1
    Cells(3, 2) = 1000
    Cells(4, 2) = 10
    Cells(5, 2) = 0.5
    Cells(6, 2) = 0.7
    Cells(7, 2) = 0.4
    Cells(8, 2) = 0.8
    Cells(9, 2) = 0.3
    Cells(10, 2) = 0.6
    Cells(11, 2) = 0.4
    Cells(12, 2) = 0.6
    Cells(13, 2) = 0.2
```

End Sub

```
Sub Ersatzproblem_Auswertung_1()
    Dim dt, tm, x, tn As Double
```

```

Dim z1, z2, z3, z4, z5, z6, z7, z8, z9 As Double
Dim t, z, zm, za As Double
Dim i, j, e, a As Long
Dim EV(3), Ez(3), AV(3), Az(3) As Long
Dim Se, Ae, Sa, Aa

```

```

dt = Cells(2, 2)
tm = Cells(3, 2)
tn = Cells(4, 2)
z1 = Cells(5, 2)
z2 = Cells(6, 2)
z3 = Cells(7, 2)
z4 = Cells(8, 2)
z5 = Cells(9, 2)
z6 = Cells(10, 2)
z7 = Cells(11, 2)
z8 = Cells(12, 2)
z9 = Cells(13, 2)

```

```

For j = 1 To 3
    Ez(j) = 0
    Az(j) = 0
Next j
Se = 0: Ae = 0
Sa = 0: Aa = 0
zm = 0
i = 2
z = Timer
Randomize
For t = dt To tm Step dt

```

```

'Einlassventile

```

```

    e = 0
    For j = 1 To 3
        z = Rnd(z)
        x = EV(j) / tn
        If z <= x Then
            e = e + 1
            Ez(j) = Ez(j) + 1
            Cells(Ez(j) + 2, 3 + j) = EV(j)
            Se = Se + EV(j)
            Ae = Ae + 1
            EV(j) = 0
        End If
    Next j

```

```

'Auslassventile

```

```

    a = 0
    For j = 1 To 3
        z = Rnd(z)
        x = AV(j) / tn
        If z <= x Then
            a = a + 1
            Az(j) = Az(j) + 1
            Cells(Az(j) + 2, 6 + j) = AV(j)
            Sa = Sa + AV(j)
            Aa = Aa + 1
            AV(j) = 0
        End If
    Next j

```

```

'Montage- und Ausfallzeiten

```

```

    If e > 0 Or a > 0 Then
        zm = zm + z1 + z9
        za = za + z1 + z2 + z3 + z4 + z5 + z6 + z7 + _
            z8 + z9
    End If

```

```

    If e > 0 Then
        zm = zm + z2 + z3 + z7 + z8
        zm = zm + e * (z4 + z5 + z6)
    End If

```

```

    If a > 0 Then
        zm = zm + z2 + z3 + z7 + z8
        zm = zm + a * (z4 + z5 + z6)
    End If

```

```
'Zeitschritt
  For j = 1 To 3
    EV(j) = EV(j) + dt
    AV(j) = AV(j) + dt
  Next j
Next t
Cells(16, 2) = zm
Cells(17, 2) = za
Cells(18, 2) = Se / Ae
Cells(19, 2) = Sa / Aa
End Sub
```

```

Sub Ersatzproblem_Auswertung_2()
Dim dt, tm, x, tn As Double
Dim z1, z2, z3, z4, z5, z6, z7, z8, z9 As Double
Dim t, z, zm, za As Double
Dim i, j, k, e, a As Long
Dim EV(3), Ez(3), AV(3), Az(3) As Long
Dim Se, Ae, Sa, Aa

dt = Cells(2, 2)
tm = Cells(3, 2)
tn = Cells(4, 2)
z1 = Cells(5, 2)
z2 = Cells(6, 2)
z3 = Cells(7, 2)
z4 = Cells(8, 2)
z5 = Cells(9, 2)
z6 = Cells(10, 2)
z7 = Cells(11, 2)
z8 = Cells(12, 2)
z9 = Cells(13, 2)

For j = 1 To 3
    Ez(j) = 0
    Az(j) = 0
Next j
Se = 0: Ae = 0
Sa = 0: Aa = 0
zm = 0: za = 0
i = 2
z = Timer
Randomize
For t = dt To tm Step dt
,
'Einlassventile
    e = 0
    For j = 1 To 3
        z = Rnd(z)
        x = EV(j) / tn
        If z <= x Then
            e = e + 1
            Ez(j) = Ez(j) + 1
            Cells(Ez(j) + 2, 3 + j) = EV(j)
            Se = Se + EV(j)
            Ae = Ae + 1
            EV(j) = 0
        End If
    Next j
,
'Auslassventile
    a = 0
    For j = 1 To 3
        z = Rnd(z)
        x = AV(j) / tn
        If z <= x Then
            a = a + 1
            Az(j) = Az(j) + 1
            Cells(Az(j) + 2, 6 + j) = AV(j)
            Sa = Sa + AV(j)
            Aa = Aa + 1
            AV(j) = 0
        End If
    Next j
,
'Montage- und Ausfallzeiten
    If e > 0 Or a > 0 Then
        zm = zm + z1 + z9
        za = za + z1 + z2 + z3 + z4 + z5 + z6 + z7 + _
            z8 + z9
    End If
    If e > 0 Then
        zm = zm + z2 + z3 + z7 + z8
        zm = zm + e * (z4 + z5 + z6)
        For k = 1 To 3
            If EV(k) >= tn / 2 Then
                zm = zm + z4 + z5 + z6
                Ez(k) = Ez(k) + 1
            End If
        Next k
    End If
Next t
End Sub

```

```
        Cells(Ez(k) + 2, 3 + k) = EV(k)
        Se = Se + EV(k)
        Ae = Ae + 1
        EV(k) = 0
    End If
Next k
End If
If a > 0 Then
    zm = zm + z2 + z3 + z7 + z8
    zm = zm + a * (z4 + z5 + z6)
    For k = 1 To 3
        If AV(k) >= tn / 2 Then
            zm = zm + z4 + z5 + z6
            Az(k) = Az(k) + 1
            Cells(Az(k) + 2, 6 + k) = AV(k)
            Sa = Sa + AV(k)
            Aa = Aa + 1
            AV(k) = 0
        End If
    Next k
End If
End If
,
'Zeitschritt
For j = 1 To 3
    EV(j) = EV(j) + dt
    AV(j) = AV(j) + dt
Next j
Next t
Cells(16, 2) = zm
Cells(17, 2) = za
Cells(18, 2) = Se / Ae
Cells(19, 2) = Sa / Aa
End Sub
```

```

Sub Ersatzproblem_Auswertung_3()
  Dim dt, tm, x, tn As Double
  Dim z1, z2, z3, z4, z5, z6, z7, z8, z9 As Double
  Dim t, z, zm, za As Double
  Dim i, j, k, e, a As Long
  Dim EV(3), Ez(3), AV(3), Az(3) As Long
  Dim Se, Ae, Sa, Aa

  dt = Cells(2, 2)
  tm = Cells(3, 2)
  tn = Cells(4, 2)
  z1 = Cells(5, 2)
  z2 = Cells(6, 2)
  z3 = Cells(7, 2)
  z4 = Cells(8, 2)
  z5 = Cells(9, 2)
  z6 = Cells(10, 2)
  z7 = Cells(11, 2)
  z8 = Cells(12, 2)
  z9 = Cells(13, 2)

  For j = 1 To 3
    Ez(j) = 0
    Az(j) = 0
  Next j
  Se = 0: Ae = 0
  Sa = 0: Aa = 0
  zm = 0: za = 0
  i = 2
  z = Timer
  Randomize
  For t = dt To tm Step dt
,
'Einlassventile
  e = 0
  For j = 1 To 3
    z = Rnd(z)
    x = EV(j) / tn
    If z <= x Then
      e = e + 1
      Ez(j) = Ez(j) + 1
      Cells(Ez(j) + 2, 3 + j) = EV(j)
      Se = Se + EV(j)
      Ae = Ae + 1
      EV(j) = 0
    End If
  Next j
,
'Auslassventile
  a = 0
  For j = 1 To 3
    z = Rnd(z)
    x = AV(j) / tn
    If z <= x Then
      a = a + 1
      Az(j) = Az(j) + 1
      Cells(Az(j) + 2, 6 + j) = AV(j)
      Sa = Sa + AV(j)
      Aa = Aa + 1
      AV(j) = 0
    End If
  Next j
,
'Montage- und Ausfallzeiten
  If e > 0 Or a > 0 Then
    zm = zm + z1 + z9
    za = za + z1 + z2 + z3 + z4 + z5 + z6 + z7 + _
      z8 + z9
  End If

  If e > 0 Then
    zm = zm + z2 + z3 + z7 + z8
    For k = 1 To 3
      zm = zm + z4 + z5 + z6
      Ez(k) = Ez(k) + 1
      Cells(Ez(k) + 2, 3 + k) = EV(k)
      Se = Se + EV(k)
    
```

```
        Ae = Ae + 1
        EV(k) = 0
    Next k
End If

If a > 0 Then
    zm = zm + z2 + z3 + z7 + z8
    For k = 1 To 3
        zm = zm + z4 + z5 + z6
        Az(k) = Az(k) + 1
        Cells(Az(k) + 2, 6 + k) = AV(k)
        Sa = Sa + AV(k)
        Aa = Aa + 1
        AV(k) = 0
    Next k
End If

' Zeitschritt
For j = 1 To 3
    EV(j) = EV(j) + dt
    AV(j) = AV(j) + dt
Next j
Next t
Cells(16, 2) = zm
Cells(17, 2) = za
Cells(18, 2) = Se / Ae
Cells(19, 2) = Sa / Aa
End Sub
```


Option Explicit

Dim n, x(), j As Integer

```
Sub Permut_Start()  
    Dim i As Integer  
  
    ThisWorkbook.Worksheets("Permutationen").Cells.Clear  
    n = InputBox("Anzahl")  
    ReDim x(n)  
    j = 0  
    For i = 1 To n  
        x(i) = i  
    Next i  
    Call Permut(1)  
End Sub
```

```
Sub Permut(k As Integer)  
    Dim i, y As Integer  
  
    y = x(k)  
    For i = k To n  
        x(k) = x(i)  
        x(i) = y  
        If k < n Then  
            Call Permut(k + 1)  
        Else  
            Call Permut_Ausgabe  
        End If  
        x(i) = x(k)  
    Next i  
    x(k) = y  
End Sub
```

```
Sub Permut_Ausgabe()  
    Dim i As Integer  
  
    j = j + 1  
    For i = 1 To n  
        Cells(j, i) = x(i)  
    Next i  
End Sub
```

```
Option Explicit
```

```
Dim MyDoc As Object
Dim n, x() As Long
Dim a(), Max As Double
```

```
Sub Engpass_Leer()
    ThisWorkbook.Worksheets("Engpass").Cells.Clear
End Sub
```

```
Sub Engpass_Test()
    Cells(1, 1) = 7
    Cells(1, 2) = 5
    Cells(1, 3) = 11
    Cells(1, 4) = 13
    Cells(1, 5) = 14
    Cells(1, 6) = 9
```

```
    Cells(2, 1) = 3
    Cells(2, 2) = 6
    Cells(2, 3) = 15
    Cells(2, 4) = 6
    Cells(2, 5) = 3
    Cells(2, 6) = 11
```

```
    Cells(3, 1) = 10
    Cells(3, 2) = 6
    Cells(3, 3) = 3
    Cells(3, 4) = 1
    Cells(3, 5) = 2
    Cells(3, 6) = 7
```

```
    Cells(4, 1) = 4
    Cells(4, 2) = 7
    Cells(4, 3) = 5
    Cells(4, 4) = 1
    Cells(4, 5) = 6
    Cells(4, 6) = 8
```

```
    Cells(5, 1) = 5
    Cells(5, 2) = 7
    Cells(5, 3) = 8
    Cells(5, 4) = 3
    Cells(5, 5) = 10
    Cells(5, 6) = 6
```

```
    Cells(6, 1) = 4
    Cells(6, 2) = 5
    Cells(6, 3) = 3
    Cells(6, 4) = 6
    Cells(6, 5) = 12
    Cells(6, 6) = 9
```

```
End Sub
```

```
Sub Enpass_Start()
    Dim i, k As Long
    Set MyDoc = ThisWorkbook.Worksheets("Engpass")
    n = MyDoc.UsedRange.Rows.Count
    Max = 0
```

```
    ReDim x(n), a(n, n)
```

```
    For i = 1 To n
        x(i) = i
        For k = 1 To n
            a(i, k) = Cells(i, k)
        Next k
    Next i
```

```
    Call Permut(1)
```

```
End Sub
```

```
Sub Permut(k As Integer)
    Dim i, y As Long
```

```
    y = x(k)
    For i = k To n
```

```
x(k) = x(i)
x(i) = y
If k < n Then
    Call Permut(k + 1)
Else
    Call Permut_Ausgabe
End If
x(i) = x(k)
Next i
x(k) = y
End Sub
```

```
Sub Permut_Ausgabe()
Dim i As Integer
Dim z As Double

z = 0
For i = 1 To n
    z = z + a(i, x(i))
Next i

If z > Max Then
    Max = z
    For i = 1 To n
        Cells(n + 2, i) = x(i)
        Cells(n + 3, i) = a(i, x(i))
    Next i
    Cells(n + 3, n + 1) = z
End If
End Sub
```

Option Explicit

```

Sub Regression_Leer()
    ThisWorkbook.Worksheets("Regression").Cells.Clear
    Call Regression_Grafik_löschen
End Sub

Sub Regression_Test()
    Dim i As Integer
    For i = 1 To 10
        Cells(i, 1) = i
    Next i
    Cells(1, 2) = 1.57
    Cells(2, 2) = 2.22
    Cells(3, 2) = 2.03
    Cells(4, 2) = 2.41
    Cells(5, 2) = 2.35
    Cells(6, 2) = 2.38
    Cells(7, 2) = 2.58
    Cells(8, 2) = 2.49
    Cells(9, 2) = 2.78
    Cells(10, 2) = 2.92
End Sub

Sub Regression_Start()
    Dim myDoc As Object
    Dim i As Long
    Dim a, b, r, sx, sy, xq, yq As Double
    Dim sxx, syy, sx2, sy2, sxy As Double
    Set MyDoc = ThisWorkbook.Worksheets("Regression")
    n = MyDoc.UsedRange.Rows.Count

    ReDim x(n), y(n)

    For i = 1 To n
        x(i) = Cells(i, 1)
        y(i) = Cells(i, 2)
        sx = sx + x(i)
        sy = sy + y(i)
    Next i
    xq = sx / n
    yq = sy / n
    For i = 1 To n
        sxx = sxx + (x(i) - xq)
        syy = syy + (y(i) - yq)
        sx2 = sx2 + (x(i) - xq) ^ 2
        sy2 = sy2 + (y(i) - yq) ^ 2
        sxy = sxy + (x(i) - xq) * (y(i) - yq)
    Next i
    a = sxy / sx2
    b = yq - a * xq
    r = sxy / Sqr(sx2) / Sqr(sy2)
    Cells(1, 5) = a
    Cells(2, 5) = b
    Cells(3, 5) = r
    For i = 1 To n
        Cells(i, 3) = a * x(i) + b
    Next i
End Sub

Sub Regression_Grafik()
    Charts.Add
    ActiveChart.ChartType = xlXYScatterSmoothNoMarkers
    ActiveChart.SetSourceData Source:= _
        Sheets("Regression").Range("C15")
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection.NewSeries
    ActiveChart.SeriesCollection(1).XValues = _
        "=Regression!R1C1:R10C1"
    ActiveChart.SeriesCollection(1).Values = _
        "=Regression!R1C2:R10C2"
    ActiveChart.SeriesCollection(1).Name = "=" & "Messwerte" & ""
    ActiveChart.SeriesCollection(2).XValues = _
        "=Regression!R1C1:R10C1"
    ActiveChart.SeriesCollection(2).Values = _
        "=Regression!R1C3:R10C3"
    ActiveChart.SeriesCollection(2).Name = _
        "=" & "Regressionsgerade" & ""

```

```
ActiveChart.Location Where:=xlLocationAsObject, _
    Name:="Regression"
ActiveChart.Axes(xlValue).Select
With ActiveChart.Axes(xlValue)
    .MinimumScale = 1.5
    .MaximumScale = 3
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
ActiveChart.Axes(xlCategory).Select
With ActiveChart.Axes(xlCategory)
    .MinimumScaleIsAuto = True
    .MaximumScale = 10
    .MinorUnitIsAuto = True
    .MajorUnitIsAuto = True
    .Crosses = xlAutomatic
    .ReversePlotOrder = False
    .ScaleType = xlLinear
    .DisplayUnit = xlNone
End With
End Sub
Sub Regression_Grafik_löschen()
    Dim Shp As Shape
    For Each Shp In Worksheets("Regression").Shapes
        Shp.Delete
    Next
End Sub
```

```
Option Explicit
Dim m, n As Integer

Sub Nutzwert_Start()
m = 0
n = 0
Call Nutzwert_Formblatt(m, n)
End Sub
```

```
Sub Nutzwert_Formblatt(m, n)
Dim i, j As Integer
Dim r, c As String

ThisWorkbook.Worksheets("Nutzwertanalyse").Cells.Clear

'Eingaben erforderlich?
If m = 0 Then
m = InputBox("Anzahl der Lösungen")
End If
If n = 0 Then
n = InputBox("Anzahl der Kriterien")
End If
If m = 0 Or n = 0 Then
MsgBox "Eingabefehler!"
Exit Sub
End If
```

```
'Spaltenbeschriftung
Columns("A:A").ColumnWidth = 30
Range("A1").Value = "Lösungen"
For i = 1 To n
c = Chr(64 + i * 2) & "1"
Range(c).Select
With Selection
.Value = Str(i) & ". Krit."
End With
Next i
```

```
'Spaltenbreite
For i = 1 To n
c = Chr(64 + i * 2) & ":" & Chr(64 + i * 2 + 1)
Columns(c).Select
With Selection
.ColumnWidth = 6
End With
Next i
```

```
c = Chr(64 + (n + 1) * 2) & "1"
Range(c).Value = "Nutzen"
```

```
'Farbkennung
For i = 1 To n
r = Chr(64 + i * 2 + 1) & "2:" & _
Chr(64 + i * 2 + 1) & LTrim(Str(1 + m))
Range(r).Select
With Selection.Interior
.ColorIndex = 15
.Pattern = xlSolid
End With
Next i
```

```
r = Chr(64 + n * 2 + 2) & "2:" & _
Chr(64 + n * 2 + 2) & LTrim(Str(1 + m))
Range(r).Select
With Selection.Interior
.ColorIndex = 15
.Pattern = xlSolid
End With
```

```
'Formeln setzen
For i = 1 To n
c = Chr(64 + i * 2 + 1)
For j = 1 To m
r = c & LTrim(Str(1 + j))
Range(r).Select
ActiveCell.FormulaR1C1 = "=RC[-1]*R[-" & _
LTrim(Str(j)) & "]C"
```

```
Next j
Next i

'Summen bestimmen
For j = 1 To m
  r = Chr(64 + n * 2 + 2) & LTrim(Str(1 + j))
  c = "="
  For i = 1 To n
    c = c + "RC[-" & LTrim(Str(2 * i - 1)) & "]"
    If i < n Then c = c & "+"
  Next i
  Range(r).Select
  ActiveCell.FormulaR1C1 = c
Next j
End Sub

Sub Nutzwert_Beispiel()
  m = 3
  n = 3
  Call Nutzwert_Formblatt(m, n)

  Cells(2, 1) = "A"
  Cells(3, 1) = "B"
  Cells(4, 1) = "C"

  Cells(1, 3) = 2
  Cells(1, 5) = 4
  Cells(1, 7) = 3

  Cells(2, 2) = 3
  Cells(3, 2) = 3
  Cells(4, 2) = 2

  Cells(2, 4) = 2
  Cells(3, 4) = 3
  Cells(4, 4) = 2

  Cells(2, 6) = 4
  Cells(3, 6) = 2
  Cells(4, 6) = 1
End Sub
```

```
Sub Quicksort_Test()  
    Dim x As Double  
    Dim i As Integer  
  
    x = Timer  
    Randomize (x)  
    For i = 1 To 100  
        Cells(i, 1) = Rnd(x)  
    Next i  
End Sub  
  
Sub Quicksort_Start()  
    Dim MyDoc As Object  
    Dim von, bis As Integer  
  
    Set MyDoc = ThisWorkbook.Worksheets("Quicksort")  
    von = 1  
    bis = MyDoc.UsedRange.Rows.Count  
    Call QuickSort(von, bis)  
End Sub  
  
Sub QuickSort(v, b)  
    Dim i, j, p As Integer  
    Dim x, y As Double  
  
    i = v  
    j = b  
    p = Int((v + b) / 2)  
    x = Cells(p, 1)  
  
    'Array aufteilen  
    Do  
        While (Cells(i, 1) < x)  
            i = i + 1  
        Wend  
        While (Cells(j, 1) > x)  
            j = j - 1  
        Wend  
  
        If i <= j Then  
            'tauschen  
            y = Cells(i, 1)  
            Cells(i, 1) = Cells(j, 1)  
            Cells(j, 1) = y  
            i = i + 1  
            j = j - 1  
        End If  
    Loop Until i > j  
  
    ' rekursive Aufrufe  
    If v < j Then  
        Call QuickSort(v, j)  
    End If  
    If i < b Then  
        Call QuickSort(i, b)  
    End If  
End Sub
```



```
Option Explicit
Dim Blatt As Worksheet
```

```
Sub Stückliste_Neu()
Dim Name As String
Dim Index As Integer

Name = InputBox("Stücklisten-Kennung eingeben!")

'Überprüfung, ob Stückliste vorhanden
For Each Blatt In Sheets
    If Name = Blatt.Name Then
        MsgBox "Stückliste bereits vorhanden", vbOKOnly
        Exit Sub
    End If
Next

'Stückliste neu erzeugen
Set Blatt = Worksheets.Add
Blatt.Name = Name
Call Stückliste_Formblatt
End Sub
```

```
Sub Stückliste_Löschen()
Dim Name, t As String

t = ThisWorkbook.ActiveSheet.Name
Name = InputBox("Zu löschende Stücklisten-Kennung eingeben!", "Löschung", t)

'Überprüfung, ob Stückliste vorhanden
For Each Blatt In Sheets
    If Name = Blatt.Name Then
        If MsgBox("Stückliste wirklich löschen?", _
            vbYesNo, "ACHTUNG!") = vbNo Then
            Exit Sub
        End If
        Blatt.Delete
        Exit Sub
    End If
Next
End Sub
```

```
Sub Stückliste_Formblatt()
Dim MyDoc As Object

Set MyDoc = ThisWorkbook.ActiveSheet
MyDoc.Activate
MyDoc.Cells.Clear

MyDoc.Range("A1") = MyDoc.Name & "- Baugruppe"
MyDoc.Range("A1:F1").Select
With Selection
    .HorizontalAlignment = xlCenter
    .VerticalAlignment = xlBottom
    .WrapText = False
    .Orientation = 0
    .AddIndent = False
    .IndentLevel = 0
    .ShrinkToFit = False
    .ReadingOrder = xlContext
    .MergeCells = True
End With
With Selection.Font
    .Name = "Arial"
    .Size = 14
    .Strikethrough = False
    .Superscript = False
    .Subscript = False
    .OutlineFont = False
    .Shadow = False
    .Underline = xlUnderlineStyleNone
    .ColorIndex = xlAutomatic
End With
Selection.Font.Bold = True

MyDoc.Range("A2") = "Stufe"
MyDoc.Range("B2") = "PosNr."
```

```

MyDoc.Range("C2") = "Menge"
MyDoc.Range("D2") = "Bezeichnung"
MyDoc.Range("E2") = "Ident-Nr."
MyDoc.Range("F2") = "Material"
MyDoc.Range("A2:A2").Select
With Selection
    .Value = "Stufe"
    .ColumnWidth = 10
End With
MyDoc.Range("B2").Select
With Selection
    .Value = "Pos.Nr."
    .ColumnWidth = 7
End With
MyDoc.Range("C2").Select
With Selection
    .Value = "Menge"
    .ColumnWidth = 8
End With
MyDoc.Range("D2").Select
With Selection
    .Value = "Bezeichnung"
    .ColumnWidth = 30
End With
MyDoc.Range("E2").Select
With Selection
    .Value = "Baugruppe"
    .ColumnWidth = 10
End With
MyDoc.Range("F2").Select
With Selection
    .Value = "Material"
    .ColumnWidth = 20
End With

```

```

MyDoc.Columns("A:A").Select
With Selection
    .NumberFormat = "@"
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlBottom
End With
MyDoc.Columns("B:C").Select
With Selection
    .NumberFormat = "0"
    .HorizontalAlignment = xlRight
    .VerticalAlignment = xlBottom
End With
MyDoc.Columns("D:F").Select
With Selection
    .NumberFormat = "@"
    .HorizontalAlignment = xlLeft
    .VerticalAlignment = xlBottom
End With

```

```

MyDoc.Range("A3").Select
End Sub

```

```

Sub Stückliste_Struktur()
    Dim MyDoc, MyDocNew As Object
    Dim Min, Max, MaxU, x As Integer
    Dim Name, t As String
    Dim i, j, k, s As Integer

    Set MyDoc = ThisWorkbook.ActiveSheet
    MyDoc.Activate

    Max = MyDoc.UsedRange.Rows.Count - 2
    Name = "S-" & MyDoc.Name

    'Überprüfung, ob Stückliste vorhanden
    For Each Blatt In Sheets
        If Name = Blatt.Name Then
            MsgBox "Stückliste bereits vorhanden", vbOKOnly
            Exit Sub
        End If
    Next

```

```

'Stückliste neu erzeugen
Set Blatt = Worksheets.Add
Blatt.Name = Name

'Mit neuer Stückliste arbeiten
Set MyDocNew = ThisWorkbook.ActiveSheet
MyDocNew.Activate
MyDocNew.Cells.Clear
Call Stückliste_Formblatt

'Übernahme der Daten
For i = 1 To Max
  For j = 1 To 6
    MyDocNew.Cells(i + 2, j) = MyDoc.Cells(i + 2, j)
  Next j
Next i

'Auflösung der Struktur
Min = 0
Do
  x = 0
  Max = MyDocNew.UsedRange.Rows.Count - 2
  For i = Min + 1 To Max
    Name = MyDocNew.Cells(i + 2, 5)
    If Not Name = "" Then
      Set MyDoc = ThisWorkbook.Worksheets(Name)
      MaxU = MyDoc.UsedRange.Rows.Count - 2
      For j = 1 To MaxU
        If Not MyDoc.Cells(j + 2, 1) = "" Then
          x = x + 1
          MyDocNew.Rows(i + 2 + x).Select
          Selection.Insert Shift:=xlDown
          MyDocNew.Cells(i + 2 + x, 1) = _
            MyDocNew.Cells(i + 2, 1) + 1
          For k = 2 To 6
            MyDocNew.Cells(i + 2 + x, k) = _
              MyDoc.Cells(j + 2, k)
          Next k
        End If
      Next j
      Min = i
      i = Max
    End If
  Next i
Loop While x > 0
'Struktur ergänzen
Max = MyDocNew.UsedRange.Rows.Count
For i = 3 To Max
  j = Val(MyDocNew.Cells(i, 1))
  If j > 1 Then
    t = ""
    For k = 2 To j
      t = t & "."
    Next k
    t = t & LTrim(Str(j))
    MyDocNew.Cells(i, 1) = t
  End If
Next i
End Sub

```

Option Explicit

Sub Netzplan_Neu()

```

ThisWorkbook.Worksheets("Netzplan").Cells.Clear

Range("A1") = "Tätigkeiten"
Range("B1") = "Optimis-" & vbLf & "tische Zeit"
Range("C1") = "Wahrschein-" & vbLf & "liche Zeit"
Range("D1") = "Pessimis-" & vbLf & "tische Zeit"
Range("F1") = "Ereignis"
Range("G1") = "Erwartete" & vbLf & "Dauer"
Range("H1") = "Abweichung"
Range("I1") = "Spätestens" & vbLf & "starten um"
Range("J1") = "Spätestens" & vbLf & "enden um"
Range("K1") = "Frühester" & vbLf & "Start um"
Range("L1") = "Spätester" & vbLf & "Start um"
Range("M1") = "Frühestes" & vbLf & "Ende um"
Range("N1") = "Spätestes" & vbLf & "Ende um"
Range("O1") = "Schlupf"
Range("P1") = "Kritischer" & vbLf & "Pfad"
Range("Q1") = "Abweichung"
Range("A1:Q1").Select
Selection.Font.Bold = True
Selection.Font.Italic = True

Columns("A:A").Select
Selection.NumberFormat = "@"
Columns("B:D").Select
Selection.NumberFormat = "#0.0"
Columns("G:Q").Select
Selection.NumberFormat = "#0.000"
Range("A2").Select

```

End Sub

Sub Netzplan_Test()

```

Cells(2, 1) = "1-2"
Cells(3, 1) = "1-3"
Cells(4, 1) = "1-4"
Cells(5, 1) = "2-5"
Cells(6, 1) = "3-6"
Cells(7, 1) = "4-6"
Cells(8, 1) = "5-7"
Cells(9, 1) = "6-7"
Cells(10, 1) = "7-8"

```

```

Cells(2, 2) = 2
Cells(3, 2) = 3
Cells(4, 2) = 4
Cells(5, 2) = 2
Cells(6, 2) = 0
Cells(7, 2) = 3
Cells(8, 2) = 1.5
Cells(9, 2) = 1
Cells(10, 2) = 2

```

```

Cells(2, 3) = 3
Cells(3, 3) = 4
Cells(4, 3) = 6
Cells(5, 3) = 4
Cells(6, 3) = 0
Cells(7, 3) = 5
Cells(8, 3) = 4
Cells(9, 3) = 3
Cells(10, 3) = 5

```

```

Cells(2, 4) = 5
Cells(3, 4) = 5
Cells(4, 4) = 9
Cells(5, 4) = 7
Cells(6, 4) = 0
Cells(7, 4) = 8.5
Cells(8, 4) = 7
Cells(9, 4) = 7
Cells(10, 4) = 9

```

End Sub

```

Sub Netzplan_Start()
Dim MyDoc As Object
Dim i, Max As Integer
Dim tx, e1, e2 As String
Dim S(), F(), a(), e() As Double
Dim t1, t2, t3 As Double
Dim V, L, SL As Double

Set MyDoc = ThisWorkbook.ActiveSheet
MyDoc.Activate

Max = MyDoc.UsedRange.Rows.Count - 1
If Max > 0 Then
    ReDim S(Max), F(Max), a(Max, 2), e(Max, 2)
    For i = 1 To Max
        tx = Cells(i + 1, 1)
        e1 = Left(tx, InStr(tx, "-") - 1)
        e2 = Right(tx, InStr(tx, "-") - 1)
        a(i, 1) = Val(e1)
        a(i, 2) = Val(e2)
        t1 = CDBl(Cells(i + 1, 2))
        t2 = CDBl(Cells(i + 1, 3))
        t3 = CDBl(Cells(i + 1, 4))
        e(i, 1) = (t1 + 4 * t2 + t3) / 6 'erwartete Dauer
        e(i, 2) = ((t3 - t1) / 6) ^ 2 'Varianz
        S(i) = 0
        F(i) = 0
    Next i

    'Auffinden der frühesten Startzeiten
    For i = 1 To Max
        If S(a(i, 2)) < S(a(i, 1)) + e(i, 1) Then
            S(a(i, 2)) = S(a(i, 1)) + e(i, 1)
        End If
    Next i
    F(a(Max, 2)) = S(a(Max, 2))

    'Auffinden der spätesten Endzeiten
    For i = Max To 1 Step -1
        If F(a(i, 1)) = 0 Or _
            F(a(i, 1)) > F(a(i, 2)) - e(i, 1) Then
            F(a(i, 1)) = F(a(i, 2)) - e(i, 1)
        End If
    Next i

    L = 0
    V = 0
    For i = 1 To Max
        SL = F(a(i, 2)) - S(a(i, 1)) - e(i, 1)
        If SL <= 0.1 Then
            Cells(i + 1, 6) = "Kritisch!"
        End If
        Cells(i + 1, 7) = e(i, 1)
        Cells(i + 1, 8) = Sqr(e(i, 2))
        If SL <= 0.1 Then
            Cells(i + 1, 9) = S(a(i, 1))
            Cells(i + 1, 10) = F(a(i, 2))
            'Aufsummierung von Pfadlänge und Varianz
            If L < F(a(i, 2)) Then L = F(a(i, 2))
            V = V + e(i, 2)
        Else
            Cells(i + 1, 11) = S(a(i, 1))
            Cells(i + 1, 12) = F(a(i, 2)) - e(i, 1)
            Cells(i + 1, 13) = S(a(i, 1)) + e(i, 1)
            Cells(i + 1, 14) = F(a(i, 2))
            Cells(i + 1, 15) = SL
        End If
    Next i
End If
Cells(2, 16) = L
Cells(2, 17) = Sqr(V)
End Sub

```

Option Explicit

```
Sub SuchListe()  
  Dim myDoc As Object  
  Dim t As String  
  Dim v, b, m, i As Long  
  
  Set myDoc = ThisWorkbook.Worksheets("Suchliste")  
  
  'Eingabe  
  t = InputBox("Suchbegriff angeben!")  
  
  'Start-Suchbereich  
  'von 1 bis Anzahl Einträge  
  v = 1  
  b = myDoc.UsedRange.Rows.Count  
  
  'Suchschleife  
  Do While b >= v  
    m = Int((v + b) / 2)  
    If t > Cells(m, 1) Then  
      v = m + 1  
    Else  
      b = m - 1  
    End If  
    i = v  
  Loop  
  
  'Ausgabe  
  Cells(i, 1).Activate  
  MsgBox "Zeile:" & Str(i) & vbLf & Cells(i, 1), _  
    vbOKOnly, "Suchergebnis von: " & t  
End Sub
```

Option Explicit

```

Sub Greedy_Neu()
    ThisWorkbook.Worksheets("Greedy").Cells.Clear

    Range("A1") = "Auf.Nr."
    Range("B1") = "Wert"
    Range("C1") = "Beleg." & vbLf & "Zeit" & vbLf & "[h]"
    Range("D1") = "Zeitraum" & vbLf & "zur Erl." & _
        vbLf & "[h]"
    Range("F1") = "Auf.Nrn."
    Range("G1") = "Ges.Wert"
    Range("A1:G1").Select
    Selection.Font.Bold = True
    Selection.Font.Italic = True
    Range("E:E").ColumnWidth = 2
    Range("A2").Select
End Sub

```

```

Sub Greedy_Test()
    Dim i As Integer

    For i = 1 To 6
        Cells(i + 1, 1) = i
    Next i
    Cells(2, 2) = 73
    Cells(3, 2) = 61
    Cells(4, 2) = 55
    Cells(5, 2) = 12
    Cells(6, 2) = 48
    Cells(7, 2) = 33
    Cells(2, 3) = 12
    Cells(3, 3) = 10
    Cells(4, 3) = 8
    Cells(5, 3) = 11
    Cells(6, 3) = 14
    Cells(7, 3) = 10
    Cells(2, 4) = 30
    Cells(3, 4) = 40
    Cells(4, 4) = 50
    Cells(5, 4) = 30
    Cells(6, 4) = 35
    Cells(7, 4) = 45
End Sub

```

```

Sub Greedy_Start()
    Dim MyDoc As Object
    Dim W, Z, wi, zi, ei As Double
    Dim i, j, k, l, wMax, kMax, n As Integer
    Dim a() As Integer
    Dim t As String

    Set MyDoc = ThisWorkbook.Worksheets("Greedy")
    n = MyDoc.UsedRange.Rows.Count - 1
    ReDim a(n, 4)

    'Daten übernehmen
    For i = 1 To n
        a(i, 2) = Cells(i + 1, 2)
        a(i, 3) = Cells(i + 1, 3)
        a(i, 4) = Cells(i + 1, 4)
    Next i

    For i = 1 To n
        'Merker löschen
        For j = 1 To n
            a(j, 1) = 0
        Next j
        'Ersten Wert setzen
        a(i, 1) = 1

        'Sortierung der Werte
        For j = 2 To n
            'grösster freier Wert
            W = 0
            l = 0
            For k = 1 To n

```

```
    If a(k, 1) = 0 Then
        If a(k, 2) > W Then
            W = a(k, 2)
            l = k
        End If
    End If
Next k
a(1, 1) = j
Next j
```

```
'Prüfung der Zulässigkeit und Ausgabe
Z = 0
W = 0
t = ""
For j = 1 To n
    For k = 1 To n
        If a(k, 1) = j Then
            If Z + a(k, 3) <= a(k, 4) Then
                Z = Z + a(k, 3)
                W = W + a(k, 2)
                If j = 1 Then
                    t = LTrim(Str(k))
                Else
                    t = t & "-" & LTrim(Str(k))
                End If
            End If
            k = n
        End If
    Next k
Next j
Cells(i + 1, 6) = t
Cells(i + 1, 7) = W
Next i
End Sub
```



```

Option Explicit
Sub EinCodes_Start()
    Dim A(16, 4), U(4), M(16) As Integer
    Dim i, j, k, l, n As Integer
    Dim p1, p2 As Integer
    Dim y, z, s As Integer
    Dim t As String
'Start
    ThisWorkbook.Worksheets("EinCodes").Cells.Clear
    Range("H:W").ColumnWidth = 1
    z = InputBox("Anzahl der Codes bis zum Stop angeben")
    y = 1

'Ausgangskonfiguration 0000
    For j = 1 To 4
        A(1, j) = 0
        Cells(1, j) = 0
    Next j

'Merker Ausgangswerte
    For i = 1 To 16
        M(i) = 0
    Next i

'Start
    i = 1

    Do

'Schritt vor
        For j = 1 To 4
            U(j) = A(i, j)
        Next j
        If M(i) < 4 Then
            M(i) = M(i) + 1
            U(M(i)) = 1 - U(M(i))

'Prüfung
            'Wird die gleiche Konstellation
            'noch einmal gefunden, ist p1=1
            p1 = 0
            For k = 1 To i
                p2 = 0
                For j = 1 To 4
                    If Not U(j) = A(k, j) Then p2 = 1
                Next j
                If p2 = 0 Then p1 = 1
            Next k
            '
            'Neue Konstellation
            If p1 = 0 Then
                i = i + 1
                For j = 1 To 4
                    A(i, j) = U(j)
                    Cells(i, j) = U(j)
                Next j

                If i = 16 Then
                    'Prüfung, ob der Code in sich geschlossen ist
                    s = 0
                    For j = 1 To 4
                        If Not A(1, j) = A(16, j) Then
                            s = s + 1
                        End If
                    Next j
                    If s = 1 Then
                        'Code brauchbar und wird registriert
                        For k = 1 To 16
                            t = ""
                            For j = 1 To 4
                                If A(k, j) = 1 Then
                                    t = t + ChrW(9608)
                                Else
                                    t = t + ChrW(8901)
                                End If
                                If j < 4 Then t = t + vbLf
                            Next j

```

```
        Cells(y, 7 + k) = t
    Next k
    y = y + 1

    'Merkerstand zeigen
    For k = 1 To 16
        Cells(k, 6) = M(k)
    Next k

    'Abbruch
    If y = z + 1 Then
        Exit Sub
    End If

    End If

    'Schritt zurück
    M(i) = 0
    i = i - 1
    End If
End If
Else
    'Schritt zurück
    M(i) = 0
    i = i - 1
    End If
Loop While i > 0
End Sub
```

Option Explicit

```
Sub Jeep_Neu()  
    ThisWorkbook.Worksheets("JeepProblem").Cells.Clear  
  
    Range("A1") = "Tankfüllung [L]"  
    Range("A2") = "Verbrauch [L/100km]"  
    Range("A3") = "Entfernung [km]"  
    Range("D1") = "Depot"  
    Range("E1") = "Entfernung"  
    Range("F1") = "Differenz"  
    Range("A1:A3").Select  
    Selection.Font.Bold = True  
    Selection.Font.Italic = True  
    Range("D1:F1").Select  
    Selection.Font.Bold = True  
    Selection.Font.Italic = True  
    Range("A:A").ColumnWidth = 20  
    Range("C:C").ColumnWidth = 5  
    Range("B1").Select  
End Sub
```

```
Sub Jeep_Test()  
    Cells(1, 2) = 80  
    Cells(2, 2) = 10  
    Cells(3, 2) = 1500  
End Sub
```

```
Sub Jeep_Start()  
    Dim v, t, e, s, u, w As Double  
    Dim i As Integer  
  
    t = Cells(1, 2)  
    v = Cells(2, 2)  
    e = Cells(3, 2)  
    i = 0  
    s = 0  
    u = 0  
  
    Do  
        i = i + 1  
        s = s + 1 / ((i - 1) * 2 + 1)  
        Cells(i + 1, 4) = i  
        w = t / v * 100 * s  
        Cells(i + 1, 5) = w  
        Cells(i + 1, 6) = w - u  
        u = w  
    Loop While t / v * 100 * s < e  
End Sub
```

```
Option Explicit
Dim Zeile As Integer
```

```
Sub Maschinen_Neu()
Dim m, p, i As Integer
Dim t As String

ThisWorkbook.Worksheets("Maschinenbelegung").Cells.Clear
Cells(1, 1) = "Folge"
Cells(1, 2) = "Wert"
Cells(1, 3) = "Pheromon"

m = InputBox("Anzahl Maschinen")
p = InputBox("Anzahl Produkte")

Cells(1, 5 + m) = "Folge"
Cells(1, 6 + m) = "Wert"
t = "A1:" & Chr(64 + 6 + m) & "1"
Range(t).Select
Selection.Font.Bold = True
Selection.Font.Italic = True
t = "D1:D" & LTrim(Str(p + 1))
Range(t).Select
Selection.Font.Bold = True
Selection.Font.Italic = True

Range("D1").Select
Selection.NumberFormat = "@"
ActiveCell.FormulaR1C1 = LTrim(Str(p)) & "/" & LTrim(Str(m))

For i = 1 To m
Cells(1, 4 + i) = "M" & LTrim(Str(i))
Next i
For i = 1 To p
Cells(1 + i, 4) = "P" & LTrim(Str(i))
Next i
End Sub
```

```
Sub Maschinen_Test()
ThisWorkbook.Worksheets("Maschinenbelegung").Cells.Clear
Range("D1").Select
Selection.NumberFormat = "@"
ActiveCell.FormulaR1C1 = "5/3"
Cells(1, 1) = "Folge"
Cells(1, 2) = "Wert"
Cells(1, 3) = "Pheromon"
Cells(1, 5 + 3) = "Folge"
Cells(1, 6 + 3) = "Wert"
Range("A1:I1").Select
Selection.Font.Bold = True
Selection.Font.Italic = True
Range("D2:D6").Select
Selection.Font.Bold = True
Selection.Font.Italic = True

Cells(2, 4) = "P1"
Cells(3, 4) = "P2"
Cells(4, 4) = "P3"
Cells(5, 4) = "P4"
Cells(6, 4) = "P5"
Cells(1, 5) = "M1"
Cells(1, 6) = "M2"
Cells(1, 7) = "M3"

Cells(2, 5) = 5
Cells(2, 6) = 3
Cells(2, 7) = 8
Cells(3, 5) = 5
Cells(3, 6) = 3
Cells(3, 7) = 4
Cells(4, 5) = 12
Cells(4, 6) = 4
Cells(4, 7) = 2
Cells(5, 5) = 4
Cells(5, 6) = 2
Cells(5, 7) = 7
Cells(6, 5) = 8
```

```
Cells(6, 6) = 2
Cells(6, 7) = 9
```

```
End Sub
```

```
Sub Maschinen_Start()
```

```
Dim g, m, n, p, q, r, v, i, j, k, l, f() As Integer
Dim t, ts, z() As String
Dim w, x, u(), s(), y(), Min As Double
```

```
Zeile = 1
x = 0
Randomize
```

```
g = InputBox("Anzahl Gruppen")
n = InputBox("Anzahl Ameisen/Gruppe")
p = Left(Cells(1, 4), InStr(Cells(1, 4), "/" ) - 1)
m = Right(Cells(1, 4), InStr(Cells(1, 4), "/" ) - 1)
```

```
ReDim z(n), y(p, 2), f(p), u(m), s(n)
```

```
For i = 1 To g          'über alle Gruppen
  For j = 1 To n        'über alle Pseudoameisen
    For k = 1 To p
      y(k, 1) = k
      y(k, 2) = 1
    Next k

    'Auswahl
    For k = 1 To p      'über alle Produkte
      Do
        q = 0

        'Berücksichtigung der Pheromonanteile
        If k > 1 And Zeile > 1 Then
          For l = 1 To p
            If y(l, 1) > 0 Then
              y(l, 2) = 1
              For r = 1 To Zeile
                ts = t & "-" & LTrim(Str(l))
                If ts = _
                  Left(Cells(r, 1), Len(ts)) Then
                  y(l, 2) = y(l, 2) + _
                    Val(Cells(r, 3))
                End If
              Next r
            End If
          Next l
        End If
      Next l
    End If

    'Wahrscheinlichkeitsverteilung
    g = 0
    For l = 1 To p
      g = g + y(l, 2)
    Next l
    x = Rnd(x)
    w = Int(x * g + 1)
    g = 0
    For l = 1 To p
      g = g + y(l, 2)
      If g >= w Then
        v = l
        l = p
      End If
    Next l
    If y(v, 1) > 0 Then
      y(v, 1) = 0
      y(v, 2) = 0
      If k = 1 Then
        t = LTrim(Str(v))
      Else
        t = t & "-" & LTrim(Str(v))
      End If
      f(k) = v
      q = 1
    End If
  Loop While q = 0
```

```

Next k
Cells(j + 1, 5 + m) = t
z(j) = t

'Bestimmung der Durchlaufzeit
For k = 1 To p
  For l = 1 To m
    If k = 1 Then
      u(l) = Cells(f(k) + 1, 1 + 4)
    Else
      If l = 1 Then
        u(l) = u(l) + Cells(f(k) + 1, 1 + 4)
      Else
        If u(l - 1) < u(l) Then
          u(l) = u(l) + _
            Cells(f(k) + 1, 1 + 4)
        Else
          u(l) = u(l - 1) + _
            Cells(f(k) + 1, 1 + 4)
        End If
      End If
    End If
  Next l
Next
Cells(j + 1, 6 + m) = u(m)
s(j) = u(m)
Next j

'Beurteilung
'Nur die minimalsten Werte werden eingetragen
For j = 1 To n
  If j = 1 Then
    Min = s(j)
  Else
    If s(j) < Min Then
      Min = s(j)
    End If
  End If
Next j
For j = 1 To n
  If s(j) = Min Then
    q = 0
    For k = 1 To Zeile
      If Cells(k, 1) = z(j) Then
        Cells(k, 3) = Cells(k, 3) + 1
        q = 1
        k = Zeile
      End If
    Next k
    If q = 0 Then
      Zeile = Zeile + 1
      Cells(Zeile, 1) = z(j)
      Cells(Zeile, 2) = s(j)
      Cells(Zeile, 3) = 1
    End If
  End If
Next j
Next i
End Sub

```

Option Explicit

Sub Träger_Optimierung()

```

Dim x1a, y1a, x1n, y1n, dx1, dy1 As Integer
Dim x2a, y2a, x2n, y2n, dx2, dy2 As Integer
Dim i, j, m1, m2, m3 As Integer
Dim x, a As Double
Dim l1a, l2a, l3a, l4a, l5a As Double
Dim l1n, l2n, l3n, l4n, l5n As Double
Dim g12a, g12n, g34a, g34n As Double

```

Randomize

For j = 1 To 50

```

x1a = 0
y1a = 0
x2a = 0
y2a = 0
a = 100
m1 = 0: m2 = 0: m3 = 0
l1a = a
l2a = a
l3a = a
l4a = a
l5a = Sqr(2) * a

```

Do

'1. Knoten

```

g12a = l1a + l2a
x = Rnd(x)
dx1 = Int(x * 3 - 1)
x = Rnd(x)
dy1 = Int(x * 3 - 1)
x1n = x1a + dx1
y1n = y1a + dy1
l1n = Sqr((a - y1n) ^ 2 + x1n ^ 2)
l2n = Sqr((a + x1n) ^ 2 + y1n ^ 2)
g12n = l1n + l2n
m1 = m1 + 1
If g12n < g12a Then
    m1 = 0
End If
Cells(1, 1) = x1a
Cells(1, 2) = y1a
Cells(1, 3) = g12a

```

'2. Knoten

```

g34a = l3a + l4a
x = Rnd(x)
dx2 = Int(x * 3 - 1)
x = Rnd(x)
dy2 = Int(x * 3 - 1)
x2n = x2a + dx2
y2n = y2a + dy2
l3n = Sqr((a - x1n + x2n) ^ 2 + (y1n - y2n) ^ 2)
l4n = Sqr((a - y2n) ^ 2 + x2n ^ 2)
g34n = l3n + l4n
m2 = m2 + 1
If g34n < g34a Then
    m2 = 0
End If
Cells(2, 1) = x2a
Cells(2, 2) = y2a
Cells(2, 3) = g34a

```

'letzte Verbindung

```

l5n = Sqr((a - x2n) ^ 2 + (a - y2n) ^ 2)
Cells(3, 3) = l5n

```

m3 = m3 + 1

If m1 = 0 And m2 = 0 Then

```

x1a = x1n
y1a = y1n
l1a = l1n
l2a = l2n
g12a = g12n
x2a = x2n

```

```
y2a = y2n  
l3a = l3n  
l4a = l4n  
g34a = g34n  
l5a = l5n  
m3 = 0
```

```
End If
```

```
Loop While m3 < 100
```

```
Cells(j, 5) = Cells(1, 1)
```

```
Cells(j, 6) = Cells(1, 2)
```

```
Cells(j, 7) = Cells(2, 1)
```

```
Cells(j, 8) = Cells(2, 2)
```

```
Cells(j, 9) = Cells(1, 3)
```

```
Cells(j, 10) = Cells(2, 3)
```

```
Cells(j, 11) = Cells(3, 3)
```

```
Cells(j, 12) = Cells(1, 3) + Cells(2, 3) + Cells(3, 3)
```

```
Next j
```

```
End Sub
```


Option Explicit

```
Sub PP_Neu()  
  ThisWorkbook.Worksheets("Packproblem").Cells.Clear  
  Cells(1, 1) = "Maximal"  
  Cells(1, 3) = "Massen:"  
  Cells(2, 3) = "Werte:"  
  Cells(4, 1) = "Masse"  
  Cells(4, 2) = "Wert"  
  Cells(4, 3) = "Codes"  
  Range("A1").Select  
  Selection.Font.Bold = True  
  Selection.Font.Italic = True  
  Range("A4:D4").Select  
  Selection.Font.Bold = True  
  Selection.Font.Italic = True  
  Range("C1:C2").Select  
  Selection.Font.Bold = True  
  Selection.Font.Italic = True  
  Range("A2").Select  
End Sub  
'Erzeugung von 50 zufallbedingten Gegenständen  
Sub PP_Test()  
  Dim i As Integer  
  Dim x, m, w As Double  
  
  Randomize  
  For i = 1 To 50  
    x = Rnd(x)  
    m = Int(x * 50) + 1    'Massen im Bereich 1...50  
    x = Rnd(x)  
    w = Int(x * 100) + 1  'Werte im Bereich 1...100  
    Cells(i + 4, 1) = m  
    Cells(i + 4, 2) = w  
  Next i  
  Cells(2, 1) = 200  
End Sub
```

```
'Erzeugung der Start-Eltern
Sub PP_Elter()
  Dim i, j, k, m, n As Integer
  Dim x, g, g1, w, w1, Max As Double

  Randomize
  Max = Cells(2, 1)
  n = ThisWorkbook.Worksheets("Packproblem"). _
    UsedRange.Rows.Count
  n = n - 4

  For k = 1 To 4
    For i = 1 To n 'Nullvektor
      Cells(i + 4, 3 + k) = 0
    Next i

    'Zufallsbedingte Auswahl von Genen
    g = 0
    w = 0
    m = 0
    Do
      x = Rnd(x)
      j = Int(x * n + 1)
      g1 = Cells(j + 4, 1)
      w1 = Cells(j + 4, 2)
      m = m + 1
      If g + g1 <= Max Then
        g = g + g1
        w = w + w1
        Cells(j + 4, 3 + k) = 1
        m = 0
      End If
    Loop While m < 50 'Abbruchkriterium
    Cells(1, 3 + k) = g
    Cells(2, 3 + k) = w
  Next k
End Sub
```

```
'Vergleich der Chromosomen
'Erfüllungskriterium maximaler Wert
Sub PP_Selection()
  Dim i, j, k, n As Integer
  Dim Max As Double
  n = ThisWorkbook.Worksheets("Packproblem"). _
    UsedRange.Rows.Count
  n = n - 4
  Max = Cells(2, 1)
  For i = 1 To 2
    If Cells(1, 5 + i) <= Max Then
      For j = 1 To 2
        If Cells(2, 5 + i) > Cells(2, 3 + j) Then
          For k = 1 To n + 4
            Cells(k, 3 + j) = Cells(k, 5 + i)
          Next k
          j = 2
          i = i + 1
        End If
      Next j
    End If
  Next i
  Range("F:G").Select
  Selection.Clear
  Range("A2").Select
End Sub
```

```
'Kreuzung nach dem Ein-Punkt-Überkreuzverfahren
```

```
'(one-point-crossover)
```

```
Sub PP_Crossover()
```

```
Dim i, j, n As Integer
```

```
Dim x, m, m1, w, w1 As Double
```

```
Randomize
```

```
n = ThisWorkbook.Worksheets("Packproblem"). _  
    UsedRange.Rows.Count
```

```
n = n - 4
```

```
x = Rnd(x)
```

```
j = Int(x * n + 1) 'Vertauschungspunkt der Chromosomen
```

```
For i = 1 To j
```

```
    Cells(i + 4, 6) = Cells(i + 4, 4)
```

```
    Cells(i + 4, 7) = Cells(i + 4, 5)
```

```
Next i
```

```
For i = j To n
```

```
    Cells(i + 4, 6) = Cells(i + 4, 5)
```

```
    Cells(i + 4, 7) = Cells(i + 4, 4)
```

```
Next i
```

```
'Ermittlung der neuen Werte
```

```
For i = 1 To 2
```

```
    m = 0
```

```
    w = 0
```

```
    For j = 1 To n
```

```
        If Cells(j + 4, 5 + i) = 1 Then
```

```
            m1 = Cells(j + 4, 1)
```

```
            w1 = Cells(j + 4, 2)
```

```
            m = m + m1
```

```
            w = w + w1
```

```
        End If
```

```
    Next j
```

```
    Cells(1, 5 + i) = m
```

```
    Cells(2, 5 + i) = w
```

```
Next i
```

```
End Sub
```

```
Sub PP_Mutation()  
  Dim i, j, n As Integer  
  Dim x, m, m1, w, w1 As Double  
  
  Randomize  
  n = ThisWorkbook.Worksheets("Packproblem"). _  
    UsedRange.Rows.Count  
  n = n - 4  
  
  For i = 1 To 2  
    x = Rnd(x)  
    If x < 0.2 Then '20% Wahrscheinlichkeit einer Mutation  
      x = Rnd(x)  
      j = Int(x * n + 1)  
      If Cells(j + 4, 5 + i) = 0 Then  
        Cells(j + 4, 5 + i) = 1  
      Else  
        Cells(j + 4, 5 + i) = 0  
      End If  
    End If  
  Next i  
  'Ermittlung der neuen Werte  
  For i = 1 To 2  
    m = 0  
    w = 0  
    For j = 1 To n  
      If Cells(j + 4, 5 + i) = 1 Then  
        m1 = Cells(j + 4, 1)  
        w1 = Cells(j + 4, 2)  
        m = m + m1  
        w = w + w1  
      End If  
    Next j  
    Cells(1, 5 + i) = m  
    Cells(2, 5 + i) = w  
  Next i  
End Sub
```

```
Sub PP_Populationen()  
  Dim i, n As Integer  
  
  n = InputBox("Anzahl der Populationen eingeben!")  
  
  For i = 1 To n  
    Call PP_Selection  
    Call PP_Crossover  
    Call PP_Mutation  
  Next i  
End Sub
```

```

Option Explicit
Option Explicit
Dim wT(4), wD(3) As Double

Sub Auswertung()
    Dim T, M(3, 5), v(4) As Double
    Dim i, i1, i2, z As Integer
    Dim u, u1, u2, sv As Double
    Dim MyDoc As Object
    Dim Shp As Shape
    Set MyDoc = ThisWorkbook.Worksheets("Fuzzy")
    MyDoc.Activate
    MyDoc.Cells.Clear
    '
    'alle Charts löschen
    For Each Shp In MyDoc.Shapes
        Shp.Delete
    Next
    '
    'Auswertungsstart
    z = 0
    '
    'Über alle Temperaturen
    For T = 600 To 1000 Step 50
        '
        'Fuzzyifizierung
        Call FuzzyTemperatur(T)
        For i = 0 To 2
            M(0, i + 1) = wT(i)
        Next i
        z = z + 1
        Cells(1, z) = T
        '
        'Über alle Drücke
        For i = 1 To 9
            Call FuzzyDrücke(i)
            For i1 = 0 To 2
                M(i1 + 1, 0) = wD(i1)
            Next i1
            '
            'Inferenzen
            For i1 = 1 To 3
                For i2 = 1 To 5
                    M(i1, i2) = Minimum(M(i1, 0), M(0, i2))
                Next i2
            Next i1
            '
            'Defuzzyifizierung
            v(4) = M(1, 1) + M(1, 2) + M(2, 1)
            v(3) = M(1, 3) + M(2, 2) + M(3, 1)
            v(2) = M(1, 4) + M(2, 3) + M(2, 4) + M(3, 2)
            v(1) = M(1, 5) + M(3, 3)
            v(0) = M(2, 5) + M(3, 4) + M(3, 5)
            sv = v(0) + v(1) + v(2) + v(3) + v(4)
            For i1 = 0 To 4
                If Not sv = 0 Then
                    v(i1) = v(i1) / sv
                End If
            Next i1
            u1 = v(1) ^ 2 * 0.25 + v(2) ^ 2 * 0.5 + _
                v(3) ^ 2 * 0.75 + v(4) ^ 2 * 1
            u2 = v(0) ^ 2 + v(1) ^ 2 + v(2) ^ 2 + _
                v(3) ^ 2 + v(4) ^ 2
            If Not u2 = 0 Then
                u = u1 / u2
            Else
                u = 0
            End If
            Cells(1 + i, z) = u
        Next i
    Next T
End Sub

Function Minimum(a, b) As Double
    If a <= b Then
        Minimum = a
    Else

```

```

    Minimum = b
End If
End Function

Sub FuzzyTemperatur(T)
Dim i As Integer
If T < 600 Then
    wT(0) = 1
Else
    If T <= 700 Then
        wT(0) = (700 - T) / 100
    Else
        wT(0) = 0
    End If
End If
For i = 1 To 4
    If T < 700 + (i - 1) * 100 Then
        wT(i) = 0
    Else
        If T <= 700 + i * 100 Then
            wT(i) = (T - (700 + (i - 1) * 100)) / 100
        Else
            If T <= 700 + (i + 1) * 100 Then
                wT(i) = (700 + (i + 1) * 100 - T) / 100
            Else
                wT(i) = 0
            End If
        End If
    End If
End If
Next i
If T < 900 Then
    wT(4) = 0
Else
    If T <= 1000 Then
        wT(4) = (T - 900) / 100
    Else
        wT(4) = 1
    End If
End If
End Sub

```

```

Sub FuzzyDrücke(i)
Select Case i
Case 1
    wD(0) = 1: wD(1) = 0: wD(2) = 0
Case 2
    wD(0) = 0.75: wD(1) = 0.25: wD(2) = 0
Case 3
    wD(0) = 0.5: wD(1) = 0.5: wD(2) = 0
Case 4
    wD(0) = 0.25: wD(1) = 0.75: wD(2) = 0
Case 5
    wD(0) = 0: wD(1) = 1: wD(2) = 0
Case 6
    wD(0) = 0: wD(1) = 0.75: wD(2) = 0.25
Case 7
    wD(0) = 0: wD(1) = 0.5: wD(2) = 0.5
Case 8
    wD(0) = 0: wD(1) = 0.25: wD(2) = 0.75
Case 9
    wD(0) = 0: wD(1) = 0: wD(2) = 1
End Select
End Sub

```