

```

1 Public Class GaussianaStd
2     Public a As Double 'x minima
3     Public b As Double 'x massima
4     Dim c As Double 'y minima
5     Dim d As Double 'y massima
6     Dim ymin, ymax As Double
7     Dim gr As Graphics
8     Dim FlagFunc As Boolean
9     Dim integ As IntegraleStd
10
11
12     Public Function cx(ByVal x As Double) As Integer
13         Dim cox As Integer
14         Try
15             'a = txtXMin.Text
16             'b = txtXMax.Text
17             cox = (((x - a)) / ((b - a))) * Panel1.Width 'calcolo l'ascissa
18                 sul pannello
19             Return cox
20         Catch ex As Exception
21             MsgBox(ex.Message)
22         End Try
23     End Function
24
25     Public Function cy(ByVal y As Double) As Integer
26         Dim coy As Integer
27         coy = (1 - (((y - c)) / ((d - c)))) * Panel1.Height 'calcolo
28             l'ordinata sul pannello
29         Return coy
30     End Function
31
32     Function f(ByVal x As Double) As Double
33         Dim ya As Double
34         ya = (1 / Math.Sqrt(2 * 3.14159265)) * (2.718281828 ^ -((x ^ 2) / 2))
35         'Funzione
36         Return ya
37     End Function
38
39     Sub funzione()
40         Dim x As Double
41         Dim ya As Single
42         gr = Panel1.CreateGraphics
43         Dim incr As Double
44         FlagFunc = True
45         incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2)) 'dimensione
46             incremento dell'indice
47         For x = (-5) To (5) Step incr 'da a fino a b incrementandosi di incr
48             Try
49                 ya = f(x) 'ritorna il valore della funzione su questa x
50                 'gr.DrawEllipse(Pens.Red, cx(x), cy(ya), 1, 1)
51                 gr.DrawLine(Pens.Blue, cx(x), (cy(ya)), cx(x + incr), cy(f(x +
52                     incr))) 'disegna una linea da x,y correnti a x,y del ciclo
53                     successivo
54             Catch ex As Exception
55             End Try
56         Next x
57     End Sub
58 End Class

```

```

51     Next
52     creaReticolo()
53 End Sub
54
55 Sub rangeY()
56     Dim x, min, max As Double
57     Dim ya As Single
58     Dim incr As Double
59     Dim flag1, flag2 As Boolean
60     flag1 = False
61     flag2 = False
62     Try
63         incr = Math.Abs(Math.Abs(b - a) / (Panel1.Width * 2))
64         min = -0.0001 'massimo e minimo iniziali molto vicini allo 0
65         max = 0.0001
66         If (a < b) Then
67             For x = a To b Step 1
68                 ya = f(x) 'ya assume il valore della funzione nell'ascissa x
69                 If (ya < min) And (ya > (-100)) Then 'se ya e' piu'
piccola di min e piu' grande del tetto minimo definito
come -100
70                     min = ya 'il minimo diventa ya
71                     flag1 = True
72                 End If
73                 If (ya > max) And (ya < 100) Then 'se ya e' piu' grande di
max e piu' piccolo del tetto massimo definito come 100
74                     max = ya 'il massimo diventa max
75                     flag2 = True
76                 End If
77             Next
78             c = (min - 2) 'diminuisco il minimo ottenuto di 2 e aumento il
massimo di 2 per evitare di avere la funzione attaccata a
punti non compresi o compresi al pelo
79             d = (max + 2)
80             If flag1 = False Then 'se non ho mai cambiato il minimo
81                 min = ya 'min diventa l'ultima ya trovata che sara' > 0
82                 For x = a To b Step 1
83                     ya = f(x) 'ya assume il valore della variabile
dipendente di x
84                     If (ya < min) Then 'se ya e' minore dell'ultimo min
posto (l'ultima ya in partenza)
85                         min = ya 'allora min diventa la corrente ya
86                     End If
87                 Next
88                 c = min - 2 'la c (valore minimo dell'ordinata della
funzione) viene abbassato di 2 per rendere visibile la
funzione
89             End If
90             If flag2 = False Then 'stesso lavoro di sopra, ma al contrario
per ottenere l'ordinata massima nel caso l'ordinata massima
sia minore di 0
91                 max = ya
92                 For x = a To b Step 1
93                     ya = f(x)
94                     If (ya > max) Then

```

```
95             max = ya
96             End If
97         Next
98         d = max + 2
99     End If
100 Else
101     MsgBox("XMin deve essere minore di Xmax!")
102 End If
103 Catch ex As Exception
104     MsgBox(ex.Message)
105 End Try
106 End Sub
107
108 Sub assi()
109     Dim gr As Graphics
110     gr = Panel1.CreateGraphics
111     gr.Clear(Color.White)
112     Try
113         If (b > 0) And (a < 0) Then
114             gr.DrawLine(Pens.Black, cx(0), (cy(c)), cx(0), (cy(d))) 'asse y
115         End If
116         If (d > 0) And (c < 0) Then
117             gr.DrawLine(Pens.Black, cx(a), cy(0), cx(b), cy(0)) 'asse x
118         End If
119         creaReticolo()
120     Catch ex As Exception
121     End Try
122 End Sub
123
124
125 Private Sub BtnDisegna_Click(ByVal sender As System.Object, ByVal e As
126     System.EventArgs) Handles BtnDisegna.Click
127     a = txtXMin.Text
128     b = txtXMax.Text
129     assi()
130     funzione()
131 End Sub
132
133 Private Sub BtnVisAssi_Click(ByVal sender As System.Object, ByVal e As
134     System.EventArgs) Handles BtnVisAssi.Click
135     If c > 0 Then 'modifico il valore degli estremi per rendere visibili
136         gli assi e ridisegno la funzione
137         c = -2
138     End If
139     If d < 0 Then
140         d = 2
141     End If
142     If a > 0 Then
143         a = -2
144     End If
145     If b < 0 Then
146         b = 2
147     End If
```

```
147     assi()
148     funzione()
149     Label3.Text = "Y Minima= " & Math.Round(c, 3) 'mostra il valore
150     Label4.Text = "Y Massima= " & Math.Round(d, 3) 'mostra il valore
151     dell'ord min troncato
152     dell'ord max troncato
153     BtnVisAssi.Visible = False
154
155 End Sub
156
157 'Private Sub BtnIndietro_Click(ByVal sender As System.Object, ByVal e As
158     System.EventArgs)
159     '    assi()
160     '    funzione()
161     '    Label3.Text = "Y Minima= " & c
162     '    Label4.Text = "Y Massima= " & d
163 'End Sub
164
165 Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
166     System.EventArgs) Handles MyBase.Load
167
168 End Sub
169
170 Public Sub creaReticolo()
171     Dim i As Single
172     Dim gr As Graphics
173     gr = Panel1.CreateGraphics
174     Dim incr As Double
175     Dim incr2 As Double
176     Dim incrY As Double
177     incr = Math.Abs(b - a) / 10
178     incr2 = cx(incr) - cx(0)
179     For i = incr2 To Panel1.Width Step incr2
180         Try
181             If (a < 0) And (b > 0) Then
182                 gr.DrawLine(Pens.Fuchsia, cx(0) + i, 0, cx(0) + i,
183                     Panel1.Height)
184                 gr.DrawLine(Pens.Fuchsia, cx(0) - i, 0, cx(0) - i,
185                     Panel1.Height)
186             Else
187                 gr.DrawLine(Pens.Fuchsia, cx(a) + i, 0, cx(a) + i,
188                     Panel1.Height)
189             End If
190         Catch ex As Exception
191
192         End Try
193     Next
194     For i = incr2 To Panel1.Height Step incr2
195         Try
196             If (c < 0) And (d > 0) Then
197                 gr.DrawLine(Pens.Fuchsia, 0, (cy(0) + i), Panel1.Width,
198                     (cy(0) + i))
199                 gr.DrawLine(Pens.Fuchsia, 0, (cy(0) - i), Panel1.Width,
200                     (cy(0) - i))
```

```

194         Else
195             gr.DrawLine(Pens.Fuchsia, 0, (cy(c) + i), Panel1.Width,
196                 (cy(c) + i))
197         End If
198     Catch ex As Exception
199     End Try
200 Next
201 lblQW.Text = "Larghezza Quadrato=" & incr
202 incrY = (Math.Abs(d - c) / (Panel1.Height / incr2))
203 lblQH.Text = "Altezza Quadrato=" & incrY
204 End Sub
205
206
207 Private Sub BtnRI_Click(ByVal sender As System.Object, ByVal e As
208     System.EventArgs) Handles BtnRI.Click
209     Dim baseRett As Double
210     Dim i As Integer
211     Dim altRett1 As Double
212     Dim altRett2 As Double
213     Dim altRett As Double
214     Dim b1, b2 As Single
215     Dim baseP As Single
216     Dim altP As Single
217     Dim n As Integer
218     If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> ""
219     Then
220         integ = New IntegraleStd(TxtN.Text)
221         lblRI.Text = "Area Rett Inf=" & Math.Round(integ.rettangoliInf(a,
222             b), 3)
223         n = TxtN.Text
224         baseRett = (b - a) / n
225         For i = 1 To n
226             Try
227                 b1 = CSng(a + (baseRett * (i - 1)))
228                 b2 = CSng(a + (baseRett * i))
229                 altRett1 = f(a + (baseRett * (i - 1)))
230                 altRett2 = f(a + (baseRett * i))
231                 baseP = cx(baseRett) - cx(0)
232                 If altRett1 < altRett2 Then
233                     altRett = altRett1
234                 Else
235                     altRett = altRett2
236                 End If
237                 altP = Math.Abs(cy(altRett) - cy(0))
238                 If ((f(altRett)) < 0) Then
239                     gr.DrawRectangle(Pens.Blue, cx(b1), cy(0), baseP,
240                         altP)
241                 Else
242                     gr.DrawRectangle(Pens.Blue, cx(b1), cy(altRett),
243                         baseP, altP)
244                 End If
245             Catch ex As Exception
246             End Try
247         Next

```

```

244     End If
245 End Sub
246
247 Private Sub BtnRS_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnRS.Click
248     Dim baseRett As Double
249     Dim i As Integer
250     Dim altRett1 As Double
251     Dim altRett2 As Double
252     Dim altRett As Double
253     Dim b1, b2 As Single
254     Dim baseP As Single
255     Dim altP As Single
256     Dim n As Integer
257     If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> "" Then
258         integ = New IntegraleStd(TxtN.Text)
259        LblRS.Text = "Area Rett Sup=" & Math.Round(integ.rettangoliSup(a, b), 3)
260         n = TxtN.Text
261         baseRett = (b - a) / n
262         For i = 1 To n
263             Try
264                 b1 = CSng(a + (baseRett * (i - 1)))
265                 b2 = CSng(a + (baseRett * (i)))
266                 altRett1 = f(a + (baseRett * (i - 1)))
267                 altRett2 = f(a + (baseRett * i))
268                 baseP = cx(baseRett) - cx(0)
269                 If altRett1 > altRett2 Then
270                     altRett = altRett1
271                 Else
272                     altRett = altRett2
273                 End If
274                 altP = Math.Abs(cy(altRett) - cy(0))
275                 If ((f(altRett)) < 0) Then
276                     gr.DrawRectangle(Pens.Green, cx(b1), cy(0), baseP, altP)
277                 Else
278                     gr.DrawRectangle(Pens.Green, cx(b1), cy(altRett), baseP, altP)
279                 End If
280             Catch ex As Exception
281             End Try
282         Next
283     End If
284 End Sub
285
286 Private Sub BtnT_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles BtnT.Click
287     Dim baseRett As Double
288     Dim i As Integer
289     Dim altRett1 As Double
290     Dim altRett2 As Double
291     Dim b1, b2 As Single
292     Dim baseP As Single

```

```

294 Dim altP1 As Single
295 Dim altP2 As Single
296 Dim n As Integer
297 Dim punto1 As Point
298 Dim punto2 As Point
299 Dim punto3 As Point
300 Dim punto4 As Point
301 If FlagFunc = True And IsNumeric(TxtN.Text) = True And TxtN.Text <> ""
302 Then
303     integ = New IntegraleStd(TxtN.Text)
304     LblT.Text = "Area Trap=" & Math.Round(integ.trapezi(a, b), 3)
305     n = TxtN.Text
306     baseRett = (b - a) / n
307     For i = 1 To n
308         Try
309             b1 = CSng(a + (baseRett * (i - 1)))
310             b2 = CSng(a + (baseRett * i))
311             altRett1 = f(a + (baseRett * (i - 1)))
312             altRett2 = f(a + (baseRett * i))
313             baseP = cx(baseRett) - cx(0)
314             altP1 = Math.Abs(cy(altRett1) - cy(0))
315             altP2 = Math.Abs(cy(altRett2) - cy(0))
316             punto1.X = cx(b1)
317             punto1.Y = cy(0)
318             punto2.X = cx(b2)
319             punto2.Y = cy(0)
320             punto3.X = cx(b2)
321             punto3.Y = cy(altRett2)
322             punto4.X = cx(b1)
323             punto4.Y = cy(altRett1)
324             gr.DrawLine(Pens.Lavender, punto2.X, punto2.Y, punto3.X,
325                 punto3.Y)
326             gr.DrawLine(Pens.IndianRed, punto3.X, punto3.Y, punto4.X,
327                 punto4.Y)
328             gr.DrawLine(Pens.IndianRed, punto4.X, punto4.Y, punto1.X,
329                 punto1.Y)
330         Catch ex As Exception
331         End Try
332     Next
333 End If
334 End Sub
335 Private Sub GaussianaStd_Shown(ByVal sender As Object, ByVal e As
336     System.EventArgs) Handles Me.Shown
337     rangeY()
338     If a < b Then
339         assi() 'disegna assi
340         funzione() 'disegna funzione
341         Label3.Text = "Y Minima= " & Math.Truncate(c) 'mostra il valore
342             dell'ord min troncato all'intero
343         Label4.Text = "Y Massima= " & Math.Truncate(d) 'mostra il valore
344             dell'ord max troncato all'intero
345     End If

```

```
343     If c > 0 Then 'casi in cui gli assi non sono visibili
344         BtnVisAssi.Visible = True 'rendo visibile un pulsante per
                                     allargare la funzione e visualizzare gli assi
345     End If
346     If d < 0 Then
347         BtnVisAssi.Visible = True
348     End If
349     If a > 0 Then
350         BtnVisAssi.Visible = True
351     End If
352     If b < 0 Then
353         BtnVisAssi.Visible = True
354     End If
355 End Sub
356 End Class
357
```