

# Relazione del programma di matematica:

Goniometria:retta,circonferenza,parabola.

## Vincoli imposti dal problema :

Disegnare una retta , una circonferenza , ed una parabola di varie dimensioni, con coordinate assegnate dall'utente.

## Specifiche integrate :

Per l'inserimento dei dati ho scelto di utilizzare delle schede, che permettono di visualizzare più componenti in uno spazio ridotto dato che sono fascicolati, cioè sovrapposti.

Così da occupare il minor spazio disponibile a beneficio del disegno.

## Struttura dati :

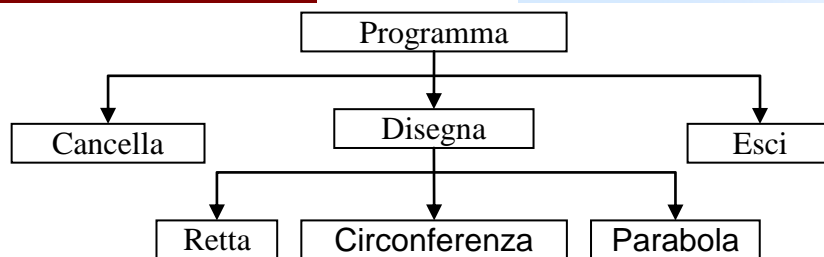
Nome	I	O	L	v/c	Descrizione	Tipo
maxf			✓	C	Costante della scala	Integer
m	✓			V	Coefficiente angolare	Single
q	✓			V	Ordinata dall'origine	Single
x1	✓			V	Ascissa del centro della circonferenza	Single
y1	✓			V	Ordinata del centro della circonferenza	Single
r	✓			V	Raggio della circonferenza	Single
A	✓			V	Coefficiente della X <sup>2</sup>	Single
B	✓			V	Coefficiente della X	Single
C	✓			V	Termine noto	Single

Queste variabili sono dichiarate nella sezione generale del form.

## Legenda

I = input, O = output, L = lavoro, v/c = variabile o costante

## Strutture del programma principale:



## Procedura : Scala assi

Questa sub routine permette di impostare la scala e di disegnare gli assi nella PICTURE BOX. Con dei tratti rossi che corrispondono ai numeri interi sull'asse. Affianco ai trattini vengono visualizzati i numeri dalla scala.

Agli esterni degli assi è visualizzata una X ed una Y per identificarli.

## Procedura : Retta

La sub routine disegna una retta utilizzando la proprietà PSet che imposta il colore di un punto, X e Y sono i valori che indicano le coordinate orizzontale (asse x) e verticale (asse y) del punto da impostare. La variabile A è calcolata in modo da restituire la prima ascissa visualizzabile.

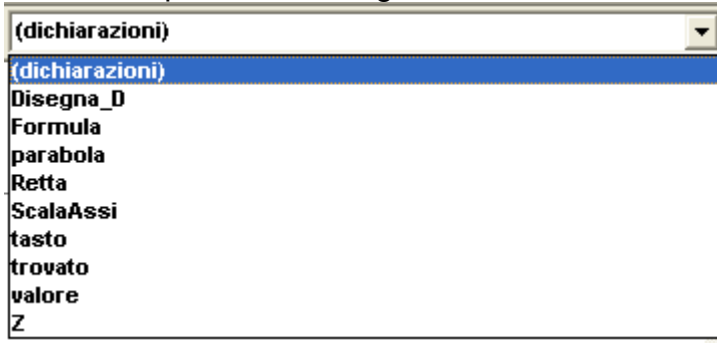
Nome	I	O	L	Descrizione	Tipo
X		✓		Incognita X	Double
Y		✓		Incognita Y	Double
A			✓	Limite visibile	Double

## Procedura : Parabola

Questa sub routine disegna una parabola con l'utilizzo di PSet. Per disegnare l'insieme dei punti bisogna considerare il

Nome	I	O	L	Descrizione	Tipo
X		✓		Incognita X	Double
Y		✓		Incognita Y	Double
inizio			✓	Inizio visibile	Double
fine			✓	Fine visibile	Double

segno di A. Nel caso è negativa, i punti vengono disegnati dall'inizio alla fine del intervallo visibile. Altrimenti A è positiva, e i punti vengono disegnati dal limite finale visibile a quello iniziale. La procedura disegna anche la direttrice, l'asse di simmetria, il fuoco e il vertice.



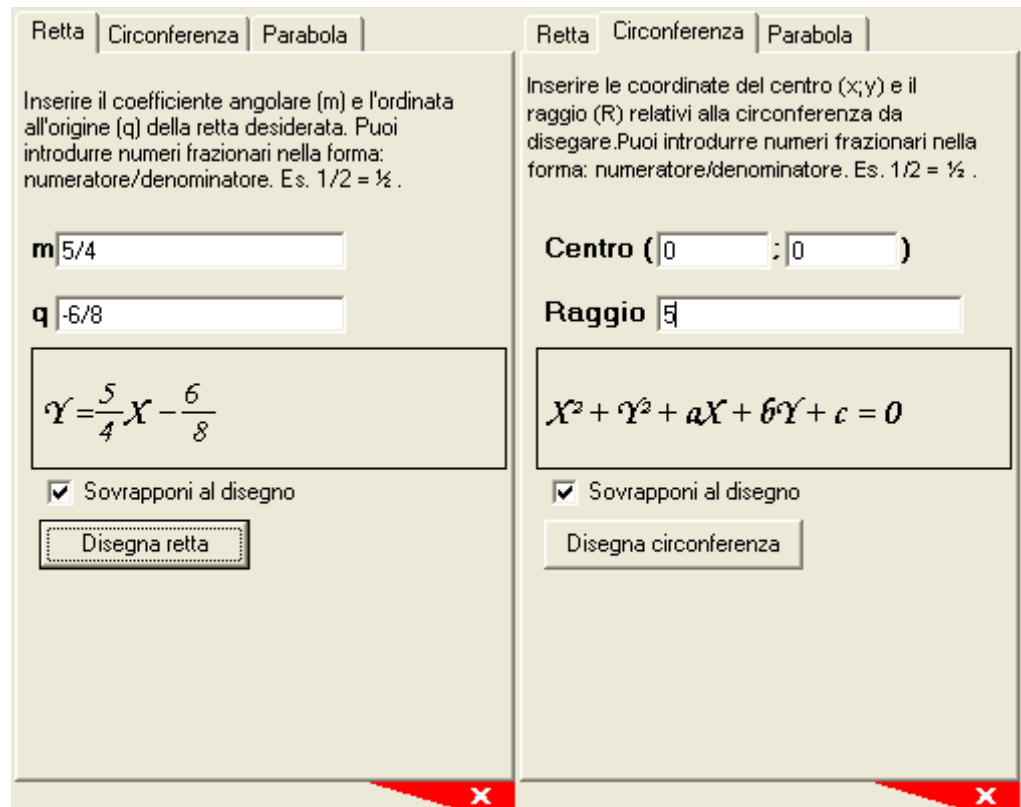
## Altre procedure

Disegna\_D: Disegna i numeri frazionari  
 Formula: Scrive le Formule matematiche.  
 Tasto: Attiva i tasti disponibili.  
 Trovato: Cerca lo slash "/" e ne restituisce la posizione.  
 Valore: Divide il numeratore per il denominatore.  
 Z: Evita gli errori della divisione per 0.

### Spiegazione del funzionamento del programma :

Tramite un editor di menù l'utente può scegliere l'opzione : cancella esci del menù "File", disegna dal menù opzioni, retta,circonferenza,parabola da sottomenù disegna. Quando si sceglie una opzione di disegno l'eventuale contenuto della PICTURE BOX viene cancellato e viene visualizzato il fascicolo di schede. che si chiama "TAB RTF" dove sarà visualizzata in

primo piano la scheda del tipo di disegno desiderato. Per visualizzare dei componenti su un controllo TAB STREP come quello che ho usato, bisogna crearsi FRAME per contenere i componenti scelti. Per semplicità ho costruito tre FRAME vettorizzate, in pratica con lo stesso nome ma con un indice diverso. In fase d'esecuzione per

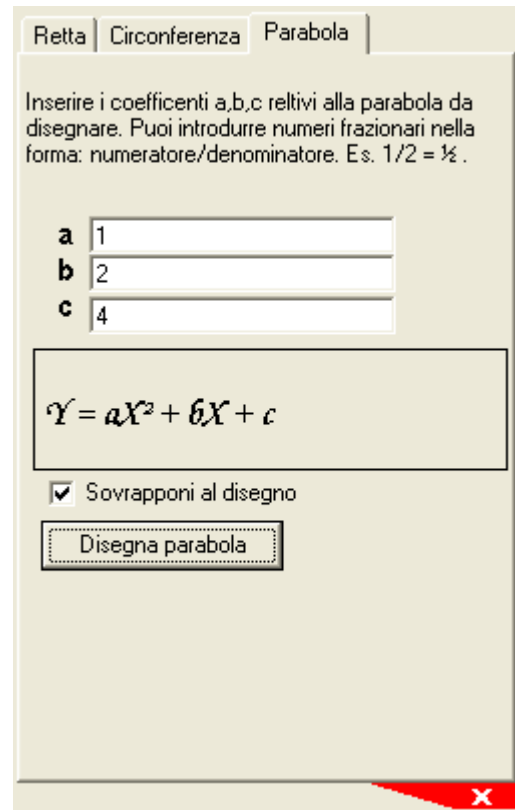


collocare le FRAME all'interno del controllo TAB STREP bisogna muovere l'altezza e la base della FRAME. Fino a quello della proprietà ClientLeft, ClientTop, ClientWidth, ClientHeight della "TAB RTF". Per visualizzare la FRAME corrispondente alla tendina cliccata, basta visualizzare la corrispondente all'indice della scheda selezionata, e nascondere tutte le altre.

Per disegnare la retta bisogna inserire il coefficiente angolare (M) e l'ordinata dall'origine (Q) ; è possibile inserire numeri frazionari sui parametri il numeratore dal denominatore con uno slash "/". La formula della retta verrà visualizzata in una PICTURE BOX, dove ho impostato la scala in modo da avere tre righe: quella centrale per il testo, quella superiore per I numeratori e quella inferiore per I denominatori. Per la circonferenza è necessaria

inserire le coordinate del centro e il raggio della circonferenza desiderata nella picture box verrà visualizzata la formula della circonferenza.

Il disegno della parabola richiede dei coefficienti a,b,c dove a non può essere uguale a 0. una volta cliccato "disegna parabola" viene visualizzata la parabola l'asse di simmetria, il fuoco e il vertice che vengono riportate in una LABEL. In tutti i e tre disegni è possibile sovrapporre il disegno a ciò che è presente sulla PICTURE BOX o disegnare solo l'ultimo luogo geometrico; questo tramite un CHECK BOX. È possibile cancellare tutto il contenuto della PICTURE BOX, per ricominciare da capo con il comando cancella nel menù file.



### Codice

```
Const maxf As Integer = 10
Dim m, q As Single
Dim x1, y1, r As Single
Dim A, B, C As Single
```

---

```
Private Sub Comcircle_Click()
```

```
  If Check1.Value = 0 Then
    Picture1.Cls
    Call ScalaAssi
```

```
  End If
```

```
  Picture1.Circle (x1, y1), r, RGB(0, 255, 0)' Il valore di colore RGB specifica
```

```
  Text6.text = "" \l'intensità relativa dei colori rosso, verde e blu che uniti
```

```
  Text7.text = "" \producono un colore specifico.
```

```
  Text8.text = ""
```

```
  x1 = 0
```

```
  y1 = 0
```

```
  r = 0
```

```
End Sub
```

---

```
Private Sub Comparabola_Click()
```

```
  If Check1.Value = 0 Then
```

```
    Picture1.Cls
    Call ScalaAssi
```

```
  End If
```

```
  Call parabola
```

```
  Text3.text = ""
```

```
  Text4.text = ""
```

```
  Text5.text = ""
```

```
End Sub
```

---

```
Private Sub ComRetta_Click()
```

```
  If Check1.Value = 0 Then
```

```
    Picture1.Cls
    Call ScalaAssi
```

```
  End If
```

```
  Call Retta
```

```
  Text1.text = ""
```

```
  Text2.text = ""
```

```
  m = 0
```

```
  q = 0
```

```
End Sub
```

---

```
Private Sub Form_Load()
```

```

' Il nome del controllo schede è "tabRTF"
' Il nome del controllo cornice è "Client..."
For i = 0 To Frame.Count - 1
    With Frame(i)
        .Move tabRTF.ClientLeft, tabRTF.ClientTop, tabRTF.ClientWidth,
tabRTF.ClientHeight
    End With
Next i
tabRTF.Visible = False
Label7.Visible = False
Check1.Visible = False
End Sub

Private Sub Frame_MouseMove(Index As Integer, Button As Integer, Shift As Integer, X As Single, Y As Single)
Label7.BackColor = &HFF&
End Sub

Private Sub Label7_Click()
Picture1.Cls
tabRTF.Visible = False
Label7.Visible = False
For i = 0 To Frame.Count - 1
    Frame(i).Visible = False
Next i
Check1.Visible = False
Label7.BackColor = &HFF&
Picture2.Visible = False
End Sub

Private Sub Label7_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label7.BackColor = &H8080FF
End Sub
Private Sub Label8_MouseMove(Button As Integer, Shift As Integer, X As Single, Y As Single)
Label7.BackColor = &HFF&
End Sub

Private Sub mnuDisegna_Click(Index As Integer)
    Picture1.Cls
    Call ScalaAssi
    tabRTF.Visible = True
    Label7.Visible = True
    Check1.Visible = True
    tabRTF.Tabs.Item(Index + 1).Selected = True
    Frame(Index).Visible = True
Select Case Index
    Case Is = 0
        Text1.SetFocus
        Call Formula(CStr("Retta"), False, False, False)
    Case Is = 1
        Text6.SetFocus
        Call Formula(CStr("circonferenza"), False, False, False)
    Case Is = 2
        Text3.SetFocus
        Call Formula(CStr("parabola"), False, False, False)
End Select
End Sub

Private Sub mnuFile_Click(Index As Integer)
Select Case Index
    Case Is = 0
        Picture1.Cls
        tabRTF.Visible = False
        Label7.Visible = False
        For i = 0 To Frame.Count - 1
            Frame(i).Visible = False

```

```

        Next i
        Picture2.Visible = False
        Check1.Visible = False
        Label7.BackColor = &HFF&
        Picture2.Visible = False
    Case Is = 2
        End
End Select
End Sub

```

---

```

Public Sub ScalaAssi()
Dim i As Integer
Picture1.Scale (-maxF, maxF)-(maxF, -maxF)
Picture1.Line (-maxF, 0)-(maxF, 0)
Picture1.Line (0, -maxF)-(0, maxF)
For i = -maxF To maxF
    Picture1.CurrentX = i
    Picture1.CurrentY = 0
    Picture1.Line (i, -0.1)-(i, 0.1), RGB(255, 0, 0)
    Picture1.CurrentX = i + 0.1
    Picture1.CurrentY = -0.01
    Picture1.Print (i)
    Picture1.CurrentX = 0
    Picture1.CurrentY = i
    Picture1.Line (-0.1, i)-(0.1, i), RGB(255, 0, 0)
    Picture1.CurrentX = -0.5
    Picture1.CurrentY = i
    If i = 0 Then
    Else
        Picture1.Print (i)
    End If
Next i
Picture1.CurrentX = 0.3
Picture1.CurrentY = maxF
Picture1.Print "X"
Picture1.CurrentX = maxF - 0.5
Picture1.CurrentY = 0.5
Picture1.Print "Y"
End Sub

```

---

```

Public Sub Retta()
Dim X, Y, A As Double
A = Abs(maxF / Z(m)) + Abs(q)
X = -A
Do
    X = X + (A / 10000)'aumenta l'intensità della retta
    Y = (m * X) + q
    Picture1.PSet (X, Y), RGB(0, 0, 250)
Loop Until X > A
End Sub

```

---

```

Private Sub tabRTF_Click()
Dim Index, i As Integer
Dim A, B, C As String
For i = 0 To Frame.Count - 1
    Frame(i).Visible = False
Next i
Index = tabRTF.SelectedItem.Index
Frame(Index - 1).Visible = True
Select Case Index
    Case 1
        Call Formula(CStr("Retta"), CBool(Text1.text <> ""), CBool(Text2.text <> ""), False)
    Case 2
        Call Formula(CStr("circonferenza"), False, False, False)

```

```

Case 3
    Call Formula(CStr("parabola"), False, False, False)
End Select
End Sub

```

---

```

Public Sub Formula(tipo As String, A, B, C As Boolean)
Dim start As Single
Picture2.Visible = True
Picture2.Scale (1, 1)-(10, 4)
Picture2.Cls
Picture2.CurrentX = 1
Picture2.CurrentY = 2
Picture2.FontName = "Monotype Corsiva"
Picture2.FontSize = 14
Select Case tipo
    Case "Retta"
        If A = False And A = True Then
            ElseIf A = True And B = True Then
                Picture2.Print " Y = "
                Call Disegna_D(2, Text1.text, start)
                Picture2.CurrentX = start
                Picture2.CurrentY = 2
                Picture2.Print "X"
                start = start + 0.5
                If Left(Text2.text, 1) = "-" Then
                    Else
                        Picture2.CurrentX = start
                        Picture2.CurrentY = 2
                        Picture2.Print "+"
                        start = start + 0.3
                    End If
                Call Disegna_D(start, Text2.text, start)
            ElseIf A = False And B = True Then
                Picture2.Print " Y = "
                start = 2
                Call Disegna_D(start, Text2.text, start)
            ElseIf A = True And B = False Then
                Picture2.Print " Y = "
                Call Disegna_D(2, Text1.text, start)
                Picture2.CurrentX = start
                Picture2.CurrentY = 2
                Picture2.Print "X"
            ElseIf A = False And B = False Then
                Picture2.Print " Y = m X + q"
            End If
        Case "circonferenza"
            Picture2.Print " X2 + Y2 + aX + bY + c = 0"
        Case "parabola"
            Picture2.Print " Y = aX2 + bX + c"
    End Select
End Sub

```

---

```

Public Function Z(ByVal Num As Single) As Single
If Num = 0 Then
    Z = 1
Else
    Z = Num
End If
End Function

```

---

```

Private Sub Text1_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 40 Or KeyCode = 13 Then
Text2.SetFocus
End If

```

```

End Sub

Private Sub Text1_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text1.text, Text1.SelStart)
End Sub

Public Function trovato(ByVal s As String) As Integer
Dim i As Integer
trovato = 0 'Inizializzazione
i = 0      'Inizializzare
Do
i = i + 1
If Mid(s, i, 1) = "/" Then
    trovato = i
End If
Loop Until i >= Len(s)
End Function

Public Function tasto(ByVal Key As Byte, ByVal tex As String, ByVal start As Byte) As Byte
Select Case Key
    Case 8, 48 To 57
        tasto = Key
    Case 44, 46
        tasto = 46
    Case 47 And trovato(tex) = 0
        tasto = Key
    Case 45 And start = 0
        tasto = Key
    Case Else
        tasto = 0
        Beep
End Select
End Function

Private Sub Text1_LostFocus()
m = valore(Text1.text)
Call Formula(CStr("Retta"), CBool(Text1.text <> ""), CBool(Text2.text <> ""), False)
End Sub

Private Sub Text2_KeyDown(KeyCode As Integer, Shift As Integer)
If KeyCode = 38 Then
    Text1.SetFocus
ElseIf KeyCode = 40 Or KeyCode = 13 Then
    ComRetta.SetFocus
End If
End Sub

Private Sub Text2_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text2.text, Text2.SelStart)
End Sub

Public Function valore(ByVal s As String) As Single
If trovato(s) > 0 Then
    valore = Val(Left(s, trovato(s) - 1)) / Val(Right(s, Len(s) - trovato(s)))
Else
    valore = Val(s)
End If
End Function

Private Sub Text2_LostFocus()
q = valore(Text2.text)
Call Formula(CStr("Retta"), CBool(Text1.text <> ""), CBool(Text2.text <> ""), False)
End Sub

Public Sub Disegna_D(ByVal startA As Single, ByVal text As String, ByRef startB As Single)
Dim Numer, Denomin, lunghezza As Single

```

```

startA = startA + 0.2
Picture2.FontSize = 13
If trovato(text) > 0 Then
    Numer = Val(Left(text, trovato(text) - 1))
    Denomin = Val(Right(text, Len(text) - trovato(text)))
    If Len(Numer) > Len(Denomin) Then
        lunghezza = (Len(Numer) / 3.75) + 0.25
        startB = startA + lunghezza
    Else
        lunghezza = (Len(Denomin) / 3.75) + 0.25
        startB = startA + lunghezza
    End If
    If Numer < 0 Then
        Picture2.Line (startA, 2.6)-(startA + 0.3, 2.6)
        startA = startA + 0.4
        startB = startB + 0.3
    End If
    Picture2.Line (startA, 2.6)-(startB, 2.6)
    Picture2.CurrentX = startA + ((lunghezza / 2)-(Len(Numer) / 3.75) / 2)- 0.1
    Picture2.CurrentY = 1.6
    Picture2.Print CStr(Abs(Numer))
    Picture2.CurrentX = startA + ((lunghezza / 2)-(Len(Denomin) / 3.75) / 2)-
0.1
    Picture2.CurrentY = 2.6
    Picture2.Print CStr(Denomin)
Else
    Picture2.CurrentX = startA
    Picture2.CurrentY = 2
    Picture2.Print text
    startB = startA + (Len(text) / 3.75)
End If
Picture2.FontSize = 14
End Sub

Private Sub Text3_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text3.text, Text3.SelStart)
End Sub

Private Sub Text3_LostFocus()
A = valore(Text3.text)
End Sub

Private Sub Text4_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text4.text, Text4.SelStart)
End Sub

Private Sub Text4_LostFocus()
B = valore(Text4.text)
End Sub

Private Sub Text5_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text5.text, Text5.SelStart)
End Sub

Private Sub Text5_LostFocus()
C = valore(Text5.text)
End Sub

Private Sub Text6_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text6.text, Text6.SelStart)
End Sub

Private Sub Text6_LostFocus()
x1 = valore(Text6.text)
End Sub

```



```

Private Sub Text7_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text7.text, Text7.SelStart)
End Sub

Private Sub Text7_LostFocus()
y1 = valore(Text7.text)
End Sub

Private Sub Text8_KeyPress(KeyAscii As Integer)
KeyAscii = tasto(KeyAscii, Text8.text, Text8.SelStart)
End Sub

Private Sub Text8_LostFocus()
r = valore(Text8.text)
End Sub

Public Sub parabola()
Dim inizio, fine, X, Y As Double
Label9.Caption = ""
If A = 0 Then
MsgBox "La parabola non esiste.", vbCritical, "Errore matematico!"
Else
'Parabola
If A < 0 Then
C = C + maxF
Else
C = C - maxF
End If
inizio = (-B - Sqr(Abs(B ^ 2 - 4 * A * C))) / (2 * A)
fine = (-B + Sqr(Abs(B ^ 2 - 4 * A * C))) / (2 * A)
If A < 0 Then
C = C - maxF
X = inizio
Do
X = X + (-inizio / 10000)
Y = A * X ^ 2 + B * X + C
Picture1.PSet (X, Y), RGB(0, 255, 255)
Loop Until X <= fine
Else
C = C + maxF
X = fine
Do
X = X + (-fine / 10000)
Y = A * X ^ 2 + B * X + C
Picture1.PSet (X, Y), RGB(0, 255, 255)
Loop Until X <= inizio
End If
'DIRETTRECE
Y = -((1 + (B ^ 2 - (4 * A * C))) / Z(4 * A))
Picture1.Line (-maxF, Y)-(maxF, Y), RGB(0, 255, 225)
Picture1.DrawStyle = 2
Picture1.Line (-maxF, Y)-(maxF, Y), RGB(0, 205, 255)
Picture1.DrawStyle = 0
Label9.Caption = "Direttrice: Y = " & CStr(Y)
'ASSE DI SIMMETRIA
X = -(B / Z(2 * A))
Picture1.Line (X, -maxF)-(X, maxF), RGB(0, 255, 225)
Picture1.DrawStyle = 2
Picture1.Line (X, -maxF)-(X, maxF), RGB(0, 205, 255)
Picture1.DrawStyle = 0
Label9.Caption = Label9.Caption & Chr(13) & "Asse di simmetria: X = " & CStr(X)
'FUOCO
Picture1.DrawWidth = 3.7
Y = (1 - (B ^ 2 - (4 * A * C))) / Z(4 * A)

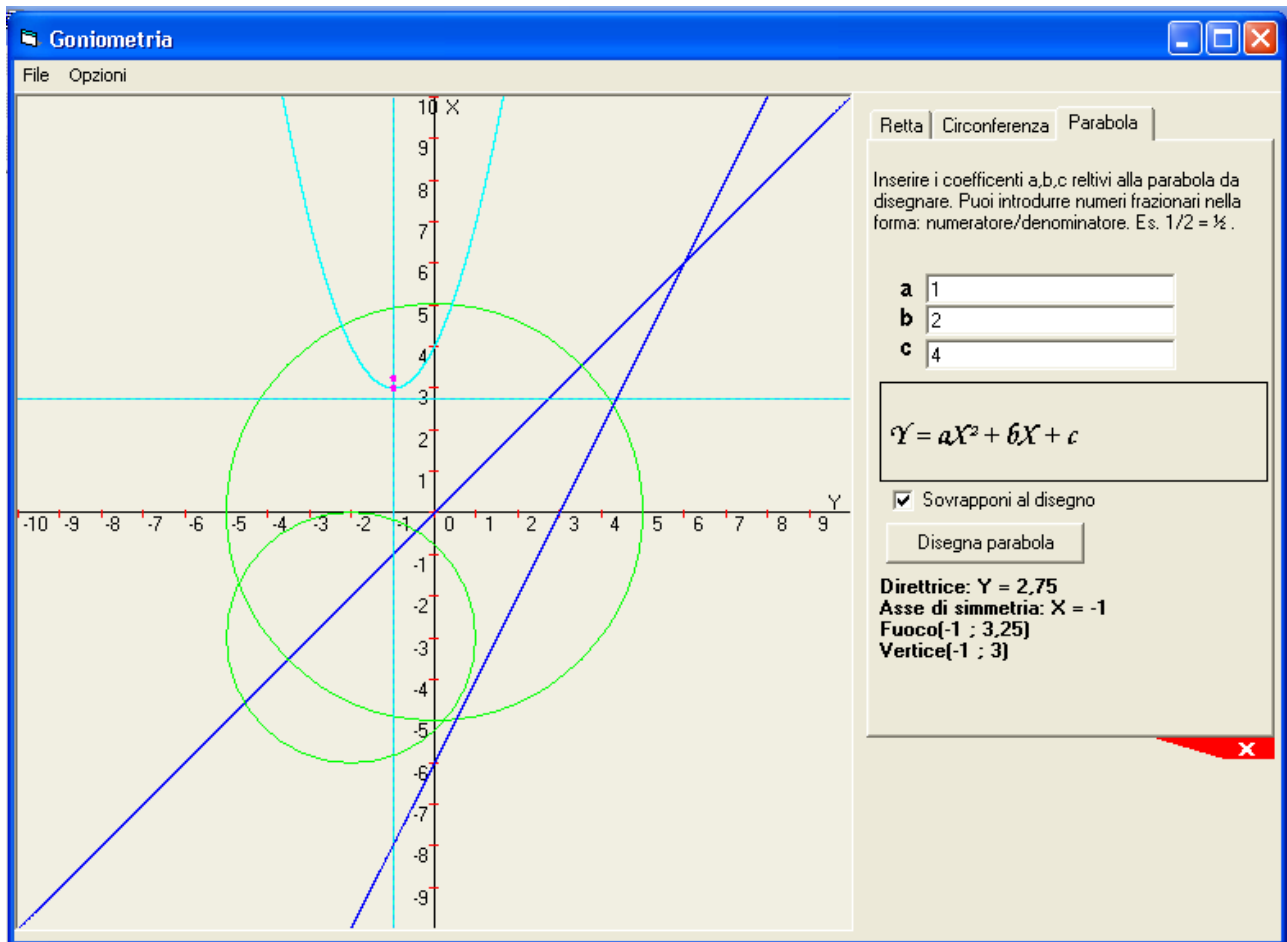
```

```

Picture1.PSet (X, Y), RGB(255, 0, 255)
Label9.Caption = Label9.Caption & Chr(13) & "Fuoco(" & CStr(X) & " ; " & CStr(Y) & ")"
'VERTICE
Y = -((B ^ 2 - (4 * A * C)) / Z(4 * A))
Picture1.PSet (X, Y), RGB(255, 0, 255)
Picture1.DrawWidth = 1
Label9.Caption = Label9.Caption & Chr(13) & "Vertice(" & CStr(X) & " ; " & CStr(Y) & ")"
End If
End Sub

```

### interfaccia:



Firma \_\_\_\_\_