

Relazione del programma di statistica:

Interpolazione tramite Lagrange

Vincoli imposti dal problema :

Realizzare un programma che permetta di determinare l'equazione del polinomio interpolante, tramite la formula di Lagrange.

Specifiche integrate :

Per memorizzare le coordinate dei punti ho utilizzato due vettori:

X() per le ascisse e Y() per ordinate. Per i coefficienti del polinomio ho usato il vettore coeff() che va da 0 a N (N è il grado del polinomio); la posizione dell'elemento nel vettore indica il grado della X.

Nella formula Lagrange il numeratore è composto di N binomi, per calcolare i coefficienti del polinomio derivato dai N binomi uso una matrice di nome combinazioni ().

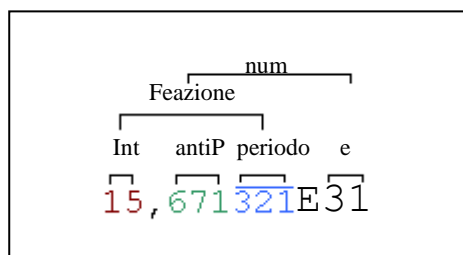
Dato che si tratta di binomi, la base è 2 quindi 2^N esprime il numero di combinazioni.

Es. $(x-1)(x-2) = (xx) + (x(-2)) + (-1x) + (-1(-2)) \rightarrow N=2 \rightarrow \text{combinazioni} = 2^2 = 4$

Per calcolare i diversi coefficienti compongo la matrice per 2N righe e N colonne, dove in ogni colonna c'è un termine noto d'ogni diverso binomio e si alternano con le X

seguendo l'andamento del codice binario. Il risultato d'ogni riga della matrice va sommato all'elemento già presente nel vettore coeff() nella posizione indicato dal numero di X presenti nella riga.

Es. $(x-1)(x-2) \rightarrow$
 combinazioni(1to 2^N , 1toN)
 $x x = (x x) = x^2$
 $x-2 = (x (-2)) = -2x$
 $-1x = (-1 x) = -x$
 $-1-2 = (-1(-2)) = 2$



Il risultato d'ogni riga della matrice va sommato all'elemento già presente nel vettore coeff() nella posizione indicato dal numero di X presenti nella riga.

Nella stampa per portare il numeratore in una frazione bisogna trovare la frazione generatrice. Separando la parte intera, la parte decimale e l'esponente, e possibile

che sia presente un numero periodico. Per questo bisogna confrontare tutti i numeri e tutti i gruppi per trovare il periodo.

In caso di presenza di periodo al numeratore si mette il numero completo letto senza virgola meno il numero intero e l'antiperiodo. Al denominatore tanti 9 quante sono le cifre del periodo e tanti 0 quante sono le cifre dell'antiperiodo.

Nel caso non ci sia il periodo al numeratore va il numero completo senza virgola e al denominatore il numero 1 con tanti 0 quante sono le cifre dell'antiperiodo o parte decimale.

Il numeratore e il denominatore vanno semplificati, per questo ho creato 2 vettori DivN() e DivD() che contengono i numeri primi derivati dalla scomposizione in fattori primi. Si pongono i numeri primi in comune a 1 e si moltiplicano. Così il risultato della moltiplicazione degli elementi di DivN() formerà il numeratore semplificato e così anche per DivD() risulterà il denominatore semplificato.

Struttura dati:

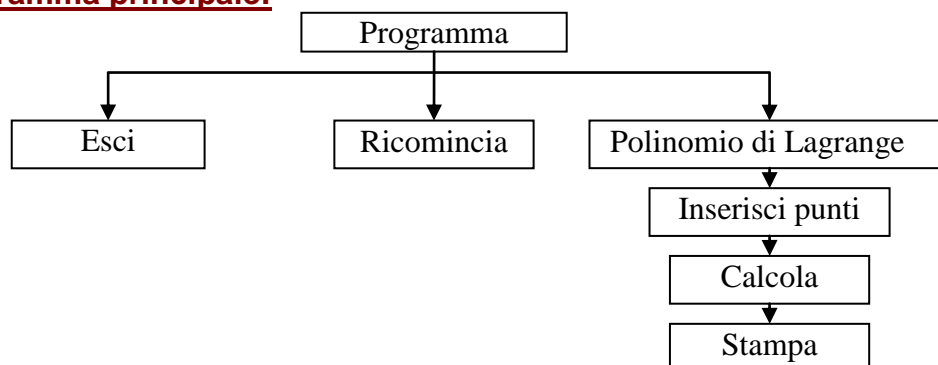
Nome	I	O	L	v/c	Descrizione	Tipo
N	✓			V	Grado del polinomio	Integer
X()	✓			V	Ascisse inserite	Single
Y()	✓			V	Ordinate inserite	Single
Coeff()		✓		V	Coefficienti del polinomio	Double
Z			✓	V	Contatore della stampa	Integer
Riga			✓	V	Contatore della riga di stampa	Single
ContatoreColonna			✓	V	Contatore della colonna di stampa	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:



Procedure :

1. **avanzamento:** Calcola lo spazio richiesto per la stampa nella PICTUREBOX.
2. **Coefficiente:** Calcola i coefficienti del polinomio tramite la matrice delle combinazioni.
3. **denominatore:** Calcola i denominatori tramite la matrice delle combinazioni
4. **Frazioni:** Trova l'eventuale frazione generatrice.
5. **Lagrange:** Calcola la matrice delle combinazioni.
6. **primo:** Calcola il primo numero primo divisore di numer.
7. **semplifica:** Riduce ai minimi termini la frazione.
8. **stampa:** Stampa la formula del polinomio.
9. **StampaFrazione:** Stampa i segni, le linee di frazione, il numeratore e il denominatore.
10. **trovato:** Trova la posizione di una stringa all'interno d'un'altra.
11. **Visualizza:** Visualizza la griglia con le coordinate del punto aggiunto.

Spiegazione del funzionamento del programma :

Per inserire un polinomio scegliere dal menù "File" "Polinomio di Lagrange" devono essere inserite le coordinate di almeno due punti, che rappresentano una retta, E' possibile inserire a piacere le coordinate d'altri punti.

Che formeranno il polinomio interpolante tramite la formula di **Lagrange**.

Per Visualizzarla cliccare sul tasto " Calcola con Lagrange" o scegliere dal menù "File" "Polinomio di Lagrange"

Nella PICTUREBOX sarà stampata la formula del polinomio.

Codice

```
Dim N As Integer  
Dim X() As Single  
Dim Y() As Single  
Dim Coeff() As Double  
Dim Z As Integer  
Dim Riga As Single, ContatoreColonna As Single
```

```

Private Sub Cmd_Inserisci_Click()
N = N + 1
ReDim Preserve X(0 To N)
ReDim Preserve Y(0 To N)
X(N) = Val(InputBox("Inserite la coordinata X del " & CStr(N + 1) & "° punto.",
"Inserisci ascissa."))
Y(N) = Val(InputBox("Inserite la coordinata Y del " & CStr(N + 1) & "° punto.",
"Inserisci ordinata."))
Call Visualizza
End Sub



---


Private Sub CmdCalcola_Click()
Call Lagrange
End Sub



---


Private Sub Form_Load()
Griglia.Visible = False
Picture1.Visible = False
Cmd_Inserisci.Visible = False
CmdCalcola.Visible = False
End Sub



---


Private Sub MnuFileSubordinato_Click(Index As Integer)
Dim risposta As Integer
Select Case Index
    Case 1
        If Griglia.Visible = False Then 'se si inizia da zero.
            MsgBox "Devono essere inserite almeno le coordinate di due punti," &
Chr(13) & " per avere un'interpolazione matematica.", vbInformation, "Devi
sapere che..."
            N = 1
            ReDim X(0 To N)
            ReDim Y(0 To N)
            For i = 0 To N
                X(i) = Val(InputBox("Inserite la coordinata X del " & CStr(i +
1) & "° punto.", "Inserisci ascissa."))
                Y(i) = Val(InputBox("Inserite la coordinata Y del " & CStr(i +
1) & "° punto.", "Inserisci ordinata."))
            Next i
            Call Visualizza
            Griglia.Visible = True
            Cmd_Inserisci.Visible = True
            CmdCalcola.Visible = True
            Labell.Visible = False
            risposta = MsgBox("È possibile calcolare il polinomio interpolante
di questi due punti." & Chr(10) & "Calcolare la formula?", vbYesNo + vbQuestion,
"Opzione.")
            If risposta = 6 Then
                Call Lagrange
            End If
        Else
            Call Lagrange
        End If
    Case 2
        Griglia.Visible = False
        Picture1.Visible = False
        Cmd_Inserisci.Visible = False
        CmdCalcola.Visible = False
        Labell.Visible = True
    Case 4
        End
End Select

```

End Sub

```
Public Sub Lagrange()
On Error GoTo Riga_Errore
Dim Msg As String
Dim Combinazioni() As String * 15
Dim stringa As String * 15
Dim i As Integer, j As Integer, m As Integer, f As Integer
Dim salto As Byte
Dim denomi As Double
Dim moltiplicando As Double
ReDim Combinazioni(1 To 2 ^ N, 1 To N) As String * 15
ReDim Coeff(0 To N)
For f = 1 To N + 1
    stringa = CStr(X(f - 1))
    GoSub Riga_C
    Call denominatore(Combinazioni(), denomi)
    moltiplicando = Y(f - 1) / denomi
    stringa = "X"
    GoSub Riga_C
    Call Coefficiente(Combinazioni(), moltiplicando)
Next f
Call stampa
Exit Sub
Riga_C:
    salto = 0
    For j = 1 To N 'colonne
        For i = 1 To 2 ^ N Step ((2 ^ N) / (2 ^ j)) * 2 'righe
            For m = i To i + ((2 ^ N) / (2 ^ j)) - 1
                Combinazioni(m, j) = stringa
                If j = f Then
                    salto = 1
                End If
                Combinazioni(m + (2 ^ N) / (2 ^ j), j) = CStr(X(j - 1 + salto) *
-1)
            Next m
        Next i
    Next j
Return
Riga_Errore:
' Verifica la presenza dell'errore, quindi visualizza il messaggio.
If Err.number <> 0 Then
    Msg = "Errore " & Str(Err.number) & " generato da " & Err.Source & Chr(13) &
"Si è verificato un errore di:" & Chr(13) & Err.Description
    MsgBox Msg, vbCritical + vbMsgBoxHelpButton, "Errore.", Err.HelpFile,
Err.HelpContext
End If
End Sub
```

```
Public Sub Visualizza()
Dim i As Integer
Griglia.Rows = N + 2
Griglia.Cols = 3
Griglia.ColWidth(0) = 350
Griglia.TextMatrix(0, 0) = "n"
Griglia.ColWidth(1) = 700
Griglia.TextMatrix(0, 1) = "X"
Griglia.ColWidth(2) = 700
Griglia.TextMatrix(0, 2) = "Y"
For i = 1 To N + 1
    Griglia.TextMatrix(i, 0) = CStr(i)
```

```

    Griglia.TextMatrix(i, 1) = X(i - 1)
    Griglia.TextMatrix(i, 2) = Y(i - 1)
Next i
End Sub

```

```

Public Sub denominatore(vett() As String * 15, Somma As Double)
Dim i As Integer, j As Integer, R As Integer, C As Integer
Dim Prodotto As Double
Somma = 0
Prodotto = 1
R = 2 ^ N 'righe
C = N 'colonne
For j = 1 To R 'righe
    For i = 1 To C 'colonne
        Prodotto = Prodotto * Val(vett(j, i))
    Next i
    Somma = Somma + Prodotto
    Prodotto = 1
Next j
End Sub

```

```

Public Sub Coefficiente(vett() As String * 15, ByVal Moltiplicatore As Double)
Dim i As Integer, j As Integer, contatoreX As Integer 'contatori
Dim R As Integer, C As Integer 'righe & colonne
Dim Somma, Prodotto As Double
Dim comodo() As Double
ReDim comodo(0 To N) As Double
Somma = 0
contatoreX = 0
Prodotto = 1
R = 2 ^ N 'righe
C = N 'colonne
For j = 1 To R 'righe
    For i = 1 To C 'colonne
        If vett(j, i) = "X" Then
            contatoreX = contatoreX + 1
        Else
            Prodotto = Prodotto * Val(vett(j, i))
        End If
    Next i
    comodo(contatoreX) = comodo(contatoreX) + Prodotto
    Prodotto = 1
    contatoreX = 0
Next j
For i = 0 To N
    comodo(i) = comodo(i) * Moltiplicatore
    Coeff(i) = Coeff(i) + comodo(i)
Next i
End Sub

```

```

Public Sub stampa()
Riga = 1
Picture1.Visible = True
Picture1.Cls
Picture1.ScaleMode = 4
Picture1.Scale (0, 0)-(50, 30)
ContatoreColonna = 0
Picture1.CurrentX = ContatoreColonna
Picture1.CurrentY = Riga
Picture1.Print "Y ="

```

```

ContatoreColonna = Picture1.TextWidth("Y =")
For Z = 0 To N
    Call Frazzioni(Abs(Coeff(Z)))
Next Z
End Sub

```

```

Public Sub Frazzioni(ByVal NumF As Double)
Dim numerat As Double, denomi As Double
Dim e As Integer, j As Integer, N As Integer, m As Integer
Dim presenzaP As Boolean
Dim frazione As String 'L'intero numero con la virgola
Dim inte As String     'Parte intera del numero
Dim antiP As String    'Parte frazionaria del numero
Dim Periodo As String  'Eventuale parte Periodica del numero
If NumF = 0 Then
    Exit Sub
End If
If trovato(CStr(NumF), ",") > 0 Then
    GoSub Riga_FrazioneGeneratrice
    Call StampFrazione(2, numerat, denomi)
Else
    numerat = NumF
    Call StampFrazione(1, numerat, denomi)
End If
Exit Sub
Riga_FrazioneGeneratrice:
e = 1
If NumF = 0 Then
    Exit Sub
End If
If trovato(CStr(NumF), "E") > 0 Then
    e = Val(Right(CStr(NumF), Len(CStr(NumF)) - trovato(CStr(NumF), "E")))
    frazione = Left(CStr(NumF), trovato(CStr(NumF), "E") - 1)
Else
    frazione = CStr(NumF)
End If
inte = Left(frazione, trovato(CStr(NumF), ",") - 1)
antiP = Right(frazione, Len(frazione) - Len(inte) - 1)
If Len(antiP) <= 7 Then
    ElseIf Mid(antiP, Len(antiP) - 2, 1) = Mid(antiP, Len(antiP) - 1, 1) And
Val(Right(antiP, 1)) - 1 = Val(Mid(antiP, Len(antiP) - 1, 1)) Then
        antiP = Left(antiP, Len(antiP) - 1) & CStr(Val(Right(antiP, 1)) - 1)
    ElseIf Mid(antiP, Len(antiP) - 3, 1) = Mid(antiP, Len(antiP) - 2, 1) And
Val(Mid(antiP, Len(antiP) - 1, 1)) + 1 = Val(Mid(antiP, Len(antiP) - 2, 1)) And
Val(Right(antiP, 1)) = 9 Then
        antiP = Left(antiP, Len(antiP) - 2) & Mid(antiP, Len(antiP) - 2, 1) &
Mid(antiP, Len(antiP) - 2, 1)
    End If
    j = 0
    Do
        j = j + 1
        Periodo = Right(antiP, Len(antiP) - j + 1)
        N = 0
        Do
            N = N + 1
            presenzaP = True
            For m = 1 To (Len(Periodo) \ N) - 1
                If Left(Periodo, N) <> Mid(Periodo, m * N - (N - 1) + 1, N) Then
                    presenzaP = False
                End If
            Next m
        Loop Until m = 2 Or presenzaP = True
    Do

```

```

Loop Until j = Len(antiP) Or presenzaP = True
If presenzaP = True Then
    Periodo = Left(Periodo, N)
    antiP = Left(antiP, j - 1)
    numerat = Val(inte & antiP & Periodo) - Val(inte & antiP)
    denomi = Val(String(Len(Periodo), "9") & String(Len(antiP), "0"))
    Call semplifica(numerat, denomi)
    If e <> 1 Then
        numerat = numerat ^ e
        denomi = denomi ^ e
        If trovato(CStr(numerat), ",") > 0 Or trovato(CStr(denomi), ",") > 0
Then
            NumF = numerat / denomi
            GoSub Riga_FrazioneGeneratrice
            End If
            Call semplifica(numerat, denomi)
        End If
    Else
        numerat = Val(inte & antiP)
        denomi = Val("1" & String(Len(antiP), "0"))
        Call semplifica(numerat, denomi)
        If e <> 1 Then
            numerat = numerat ^ e
            denomi = denomi ^ e
            If trovato(CStr(numerat), ",") > 0 Or trovato(CStr(denomi), ",") > 0
Then
                NumF = numerat / denomi
                GoSub Riga_FrazioneGeneratrice
                End If
                Call semplifica(numerat, denomi)
            End If
        End If
    End If
Return
End Sub

```

```

Public Function trovato(ByVal stringa As String, ByVal carattere As String) As
Integer
Dim i As Integer
trovato = 0
i = 0
Do
i = i + 1
If Mid(stringa, i, Len(carattere)) = carattere Then
    trovato = i
End If
Loop Until trovato = i Or i = Len(stringa)
End Function

```

```

Public Sub semplifica(ByRef N As Double, ByRef D As Double)
Dim DivisoreP As Long, Pn As Integer, Pd As Integer
Dim i As Integer, j As Integer
Dim DivN() As Long
Dim DivD() As Long
Pn = 0
DivisoreP = 2
Do Until N = 1 Or N = 0
    Pn = Pn + 1
    DivisoreP = primo(DivisoreP, N)
    ReDim Preserve DivN(1 To Pn)
    DivN(Pn) = DivisoreP
    N = N / DivisoreP

```

```

Loop
Pd = 0
DivisoreP = 2
Do Until D = 1 Or D = 0
    Pd = Pd + 1
    DivisoreP = primo(DivisoreP, D)
    ReDim Preserve DivD(1 To Pd)
    DivD(Pd) = DivisoreP
    D = D / DivisoreP
Loop
For i = 1 To Pn
    j = 0
    Do
        j = j + 1
        If DivN(i) = DivD(j) Then
            DivN(i) = 1
            DivD(j) = 1
            Exit Do
        End If
    Loop Until j = Pd
Next i
For i = 1 To Pn
    N = N * DivN(i)
Next i
For i = 1 To Pd
    D = D * DivD(i)
Next i
End Sub

```

```

Public Function primo(ByVal start As Long, ByVal number As Double) As Long
Dim i As Integer
Dim pri As Boolean
Do
    If trovato(CStr(number / start), ",") > 0 Then 'Sostituisce la funzione Mod
        start = start + 1
    End If
    pri = True
    For i = 2 To Sqr(start)
        If start Mod i = 0 Then
            pri = False
            start = start + 1
            Exit For
        End If
    Next i
Loop Until (trovato(CStr(number / start), ",") = 0 And pri = True) Or start >=
number
primo = start
End Function

```

```

Public Sub StampFrazione(mode As Byte, ByVal Ns As Double, ByVal Ds As Double)
Const Space As Byte = 0.5
Dim Lunghezza As Single
If Coeff(Z) < 0 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" - ", 0)
    Picture1.CurrentY = Riga
    Picture1.Print " - "
ElseIf Coeff(Z) > 0 And Z <> 0 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" + ", 0)
    Picture1.CurrentY = Riga

```



```

    Picture1.Print " + "
ElseIf Coeff(Z) > 0 And Z = 0 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" ", 0)
    Picture1.CurrentY = Riga
    Picture1.Print " "
End If
If Abs(Coeff(Z)) = 1 And Z <> 0 Then
    mode = 3
ElseIf Coeff(Z) = 1 And Z = 0 Then
    mode = 1
End If
If mode = 1 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(CStr(Ns), 0)
    Picture1.CurrentY = Riga
    Picture1.Print Ns
ElseIf mode = 2 Then
    If Picture1.TextWidth(CStr(Ns)) > Picture1.TextWidth(CStr(Ds)) Then
        Lunghezza = Picture1.TextWidth(CStr(Ns)) + Space
    Else
        Lunghezza = Picture1.TextWidth(CStr(Ds)) + Space
    End If
    Call avanzamento("", Lunghezza)
    Picture1.Line (ContatoreColonna - Lunghezza, Riga + 0.5)-(ContatoreColonna,
Riga + 0.5)
    Picture1.CurrentX = ContatoreColonna - Lunghezza + ((Lunghezza -
Picture1.TextWidth(CStr(Ns))) / 2) - 0.2
    Picture1.CurrentY = Riga - 0.5
    Picture1.Print Ns
    Picture1.CurrentX = ContatoreColonna - Lunghezza + ((Lunghezza -
Picture1.TextWidth(CStr(Ds))) / 2) - 0.4
    Picture1.CurrentY = Riga + 0.5
    Picture1.Print Ds
End If
If Z = 1 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" X", 0)
    Picture1.CurrentY = Riga
    Picture1.Print " X"
ElseIf Z <> 1 And Z <> 0 Then
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" X", 0)
    Picture1.CurrentY = Riga
    Picture1.Print " X"
    Picture1.Font.Size = 9
    Picture1.CurrentX = ContatoreColonna
    Call avanzamento(" " & CStr(Z), 0)
    Picture1.CurrentY = Riga - 0.2
    Picture1.Print " " & CStr(Z)
    Picture1.Font.Size = 11
End If
End Sub

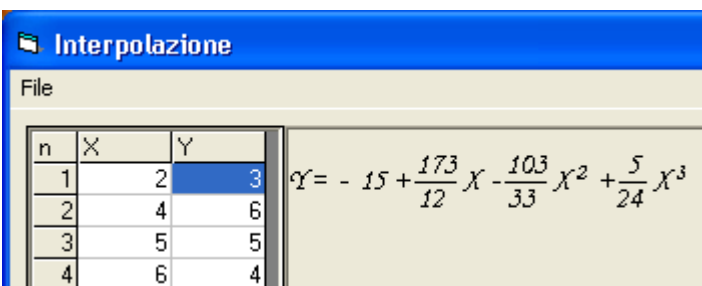
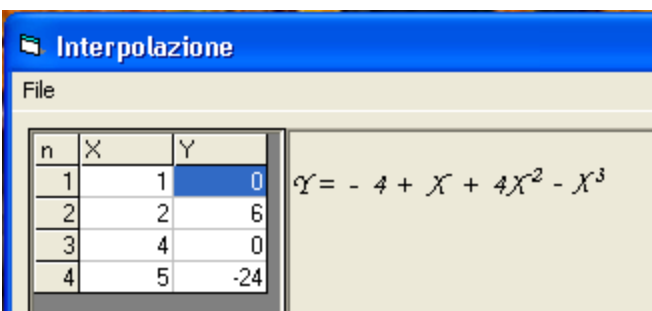
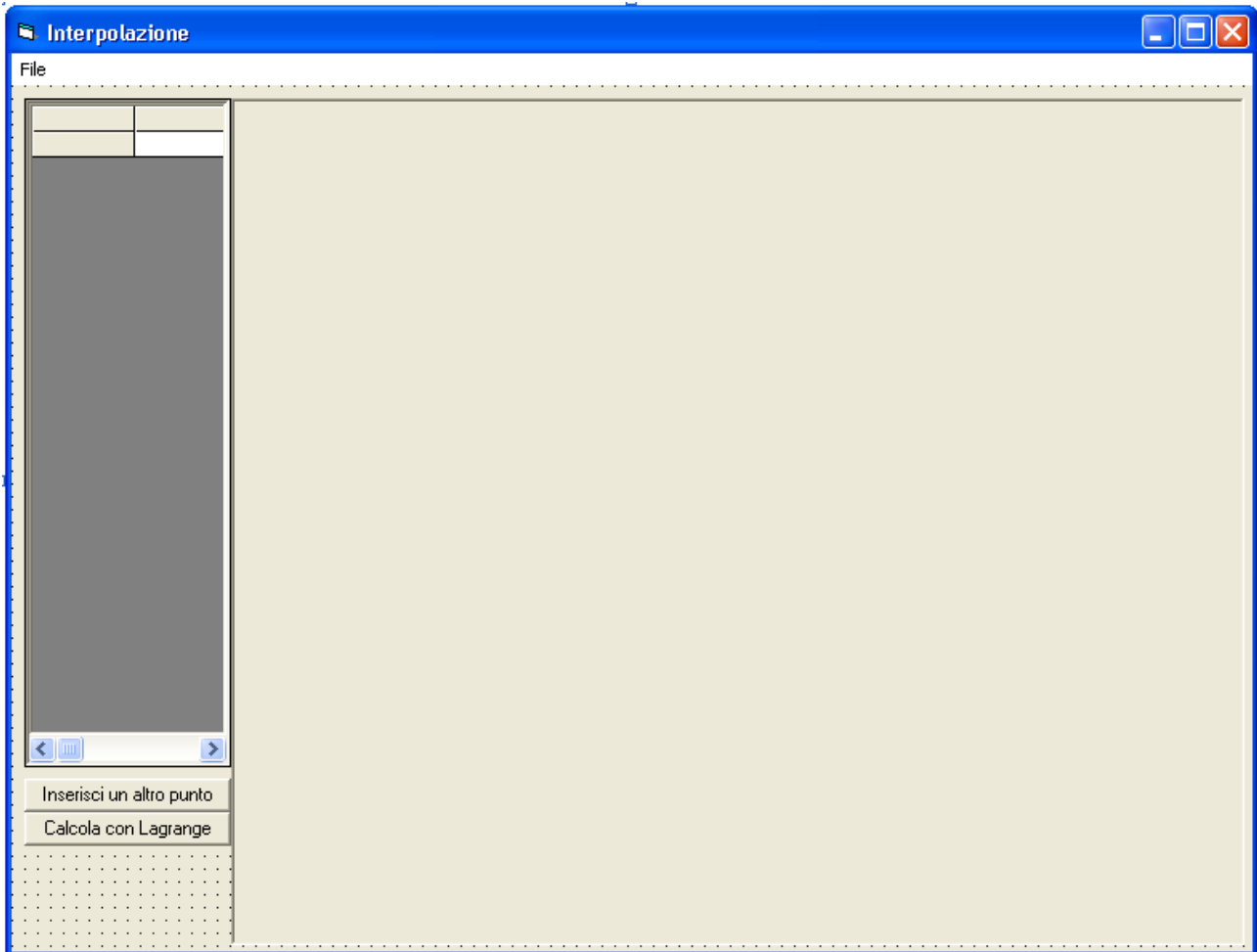
```

```

Public Function avanzamento(ByVal Testo As String, ByVal Width As Single)
If ContatoreColonna + Picture1.TextWidth(Testo) + Width > 48 Then
    Riga = Riga + 2.5
    ContatoreColonna = Picture1.TextWidth(Testo) + Width
    Picture1.CurrentX = ContatoreColonna
Else
    ContatoreColonna = ContatoreColonna + Picture1.TextWidth(Testo) + Width
End If
avanzamento = ContatoreColonna

```

interfaccia:



Interpolazione

File

n	X	Y
1	2	5
2	3	4
3	6	2
4	5	7
5	4	2


$$Y = -153 + \frac{2359}{12}X - \frac{2075}{24}X^2 + \frac{191}{12}X^3 - \frac{25}{24}X^4$$

Devi sapere che...

 Devono essere inserite almeno le coordinate di due punti, per avere un'interpolazione matematica.

OK

Errore.

 Errore 6 generato da Progetto_Interpolazione
Si è verificato un errore di:
Overflow

OK ?

Firma _____