

## Relazione del programma di statistica: Bernoulli Poisson.

### Vincoli imposti dal problema :

Predisporre un programma che, assegnati in input  $n$ ,  $p$  permetta di determinare la distribuzione della variabile casuale  $X$  con la formula di Bernoulli e quella di Poisson. Verificare che, al crescere di  $n$ , le distribuzioni tendono a sovrapporsi. In output si richiede inoltre la rappresentazione grafica delle due distribuzioni. (Es. 4 Pag.301)

### Specifiche integrate :

L'utente può inserire direttamente i valori di  $n$  e di  $p$ , questi saranno controllati e se sono compresi nell'insieme dei valori accettabili viene direttamente chiamata la sub routine, che calcola i valori memorizzandoli in vettori Double, vengono populate le tabelle e disegnati gli assi e la numerazione dei grafici, infine vengono disegnate le barre delle probabilità prima quelle di Poisson poi quelle di Bernoulli.

### Struttura dati:

Nome	I	O	L	Descrizione	Tipo
e			✓	Numero di Neplero ( $e = 2.71828182\dots$ )	Double
Lambda		✓	✓	Parametro della formula di Poisson ( $\lambda = n \cdot p$ )	Double
n	✓			Numero di prove ripetute	Integer
p	✓			Probabilità positiva dell'evento	Double
TabellaP()		✓		Vettore delle distribuzioni di Poisson	Double
TabellaB()		✓		Vettore delle distribuzioni di Bernoulli	Double
max			✓	Numero delle distribuzioni di Poisson	Integer

Queste variabili sono dichiarate nel Form1.

### **Legenda**

**I = input, O = output, L = lavoro**

### Procedure :

- 1• CalcolaTabelle:** Procedura principale per la gestione del programma è chiamata automaticamente, quando l'utente inserisce  $n$ ,  $p$  correttamente.
- 2• DisegnaAssi:** Impostala la scala delle PictureBox con un decimo di margine( $m$ ) disegna gli assi cartesiani con la propria numerazione, stampa sullo sfondo il nome del grafico in grigio, e per Poisson il valore di Lambda mentre per Bernoulli I valori di  $p$  e  $q$ .
- 3• DisegnaBernoulli:** Disegna una barra verde e spessa che rappresenta la probabilità di Bernoulli relativa all'indice  $j$  passato alla sub routine.
- 4• DisegnaPoisson:** Disegna una barra verde e spessa che rappresenta la probabilità di Poisson relativa all'indice  $j$ .
- 5• fattoriale:** Function che calcola il fattoriale, del numero Integer passato alla funzione, e lo restituisce Double.

### Spiegazione del funzionamento del programma :

Il programma ha un funzionamento molto semplice. L'utente deve solo inserire correttamente il valore di  $n$  (che può essere un numero positivo intero) e di  $p$  (che deve essere compreso tra 0 e 1). Automaticamente se i valori sono accettabili sono calcolate le tabelle e visualizzate insieme ai grafici. I valori sono tutti arrotondati a 10 cifre con la funzione Round.

**Codice:**

---

```
Const e As Double = 2.71828182845905
Dim Lambda As Double
Dim n As Integer
Dim p As Double
Dim TabellaP() As Double
Dim TabellaB() As Double
Dim max As Integer
```

---

```
Private Sub Command1_Click()
MsgBox "Grazie per aver scelto questo programma." & Chr(13) & _
" Arrivederci alla prossima.", vbInformation, "Ciao Ciao!"
End
End Sub
```

---

```
Private Sub Form_Load()
'Nasconde gli elementi vuoti
GrigliaPoisson.Visible = False
GrigliaBernoulli.Visible = False
PicturePoisson.Visible = False
PictureBernoulli.Visible = False
End Sub
```

---

```
Private Sub TextN_Change()
'Controlla la validità dell'inserimento
If Not IsNumeric(TextN.Text) Or Val(TextN.Text) < 0 Then
MsgBox "È possibile inserire solo numeri interi positivi.", vbCritical, _
"Errore! Inserire il numero di prove ripetute!"
TextN.Text = ""
Else
If TextP.Text <> "" Then
n = Val(TextN.Text)
p = Val(TextP.Text)
Call CalcolaTabelle
End If
End If
End Sub
```

---

```
Private Sub TextP_Change()
'Controlla la validità dell'inserimento
If Not IsNumeric(TextP.Text) Or Val(TextP.Text) < 0 Or Val(TextP.Text) > 1 Then
MsgBox "È possibile inserire solo numeri compresi tra 0 e 1. ES:0.5; 0.24;..." _
, vbCritical, "Errore! Inserire la probabilità positiva dell'evento!"
TextP.Text = ""
Else
If TextN.Text <> "" Then
n = Val(TextN.Text)
p = Val(TextP.Text)
Call CalcolaTabelle
End If
End If
End Sub
```

---

```
Public Sub CalcolaTabelle()
Dim x As Integer
Dim Px As Double
```

## Relazione di statistica.

```
'Visualizzazione.
GrigliaPoisson.Visible = True
GrigliaBernoulli.Visible = True
PicturePoisson.Visible = True
PictureBernoulli.Visible = True
'Calcola Tabella Poisson
Lambda = n * p
max = Int(Lambda + 4)
ReDim TabellaP(0 To max)
'Disegna tabella Poisson
GrigliaPoisson.Rows = max + 2
GrigliaPoisson.TextMatrix(0, 0) = "X"
GrigliaPoisson.TextMatrix(0, 1) = "Px"
GrigliaPoisson.ColWidth(0) = 400
GrigliaPoisson.ColWidth(1) = 1500
Call DisegnaAssi
'Popola tabella Poisson
For x = 0 To max
    Px = (Lambda ^ x / fattoriale(x)) * e ^ -Lambda
    GrigliaPoisson.TextMatrix(x + 1, 0) = CStr(x)
    GrigliaPoisson.TextMatrix(x + 1, 1) = CStr(Round(Px, 10))
    TabellaP(x) = Px
    Call DisegnaPoisson(x)
Next x
'Calcola Tabella Bernoulli
ReDim TabellaB(0 To n)
'Disegna tabella Bernoulli
GrigliaBernoulli.Rows = n + 2
GrigliaBernoulli.TextMatrix(0, 0) = "X"
GrigliaBernoulli.TextMatrix(0, 1) = "Px"
GrigliaBernoulli.ColWidth(0) = 400
GrigliaBernoulli.ColWidth(1) = 1500
'Popola tabella Bernoulli
For x = 0 To n
    Px = (fattoriale(n) / (fattoriale(x) * fattoriale(n - x))) * (p ^ x) * _
        (1 - p) ^ (n - x)
    GrigliaBernoulli.TextMatrix(x + 1, 0) = CStr(x)
    GrigliaBernoulli.TextMatrix(x + 1, 1) = CStr(Round(Px, 10))
    TabellaB(x) = Px
    Call DisegnaBernoulli(x)
Next x
End Sub
```

---

```
Public Function fattoriale(x As Integer) As Double
Dim i As Integer
fattoriale = 1
For i = 1 To x
    fattoriale = fattoriale * i
Next i
End Function
```

---

```
Public Sub DisegnaPoisson(j As Integer)
PicturePoisson.DrawWidth = 3
PicturePoisson.Line (j, 0)-(j, TabellaP(j)), RGB(0, 255, 0)
PicturePoisson.DrawWidth = 1
End Sub
```

---

```
Public Sub DisegnaAssi()
Dim Zero As Boolean 'nel caso di n = 0
Dim i, x As Double
```

## Relazione di statistica.

```
Dim m As Single 'margine
'Disegna Poisson
m = max / 10
PicturePoisson.Cls
PicturePoisson.Scale (0 - m, 1.1)-(max + m, -0.1)
'Asse Y
PicturePoisson.Line (0, 0 - m)-(0, 1 + m), RGB(0, 0, 255)
'Asse X
PicturePoisson.Line (0 - m, 0)-(max + m, 0), RGB(0, 0, 255)
'Nomi assi
PicturePoisson.CurrentX = m / 10
PicturePoisson.CurrentY = 1.1
PicturePoisson.Print "Px"
PicturePoisson.CurrentX = max + m / 2
PicturePoisson.CurrentY = 0.05
PicturePoisson.Print "X"
'Numerazione
PicturePoisson.CurrentX = -m / 2
PicturePoisson.CurrentY = -0.02
PicturePoisson.Print "o"
x = CInt(m)
If x = 0 Then x = 1
For i = 1 To max Step x
    PicturePoisson.CurrentX = i - PicturePoisson.TextWidth(CStr(i)) / 2
    PicturePoisson.CurrentY = -0.02
    PicturePoisson.Print CStr(i)
Next i
For i = 0.1 To 1 Step 0.1
    PicturePoisson.CurrentX = -m / 1.2
    PicturePoisson.CurrentY = i + 0.02
    PicturePoisson.Print CStr(i)
    PicturePoisson.Line (-m / 10, i)-(m / 10, i), RGB(0, 0, 255)
Next i
'Stampa Lambda
PicturePoisson.FontSize = 36
PicturePoisson.FontName = "Times New Roman"
PicturePoisson.ForeColor = RGB(180, 180, 180)
PicturePoisson.CurrentX = max / 2 - PicturePoisson.TextWidth("Poisson") / 2
PicturePoisson.CurrentY = 0.7
PicturePoisson.Print "Poisson"
PicturePoisson.ForeColor = vbBlack
PicturePoisson.FontSize = 18
PicturePoisson.CurrentX = max / 2 - PicturePoisson.TextWidth("l = " & _
    CStr(Lambda)) / 2
PicturePoisson.CurrentY = 1.1
PicturePoisson.FontName = "Symbol"
PicturePoisson.Print "l"
PicturePoisson.FontSize = 12
PicturePoisson.FontName = "MS Sans Serif"
PicturePoisson.CurrentX = max / 2 - PicturePoisson.TextWidth("l = " & _
    CStr(Lambda)) / 2
PicturePoisson.CurrentY = 1.09
PicturePoisson.Print " = " & CStr(Lambda)
PicturePoisson.FontSize = 8

'Disegna Bernoulli
If n = 0 Then n = 1: Zero = True
m = n / 10
PictureBernoulli.Cls
PictureBernoulli.Scale (0 - m, 1.1)-(n + m, -0.1)
'Asse Y
PictureBernoulli.Line (0, 0 - m)-(0, 1 + m), RGB(0, 0, 255)
'Asse X
```

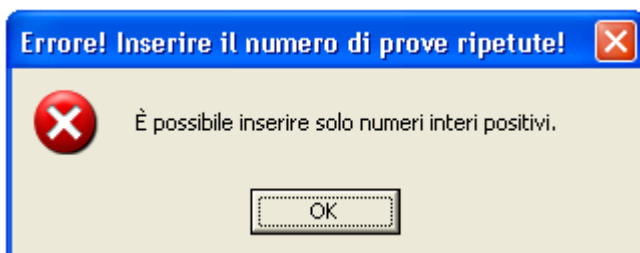
## Relazione di statistica.

```
PictureBernoulli.Line (0 - m, 0)-(n + m, 0), RGB(0, 0, 255)
'Nomi assi
PictureBernoulli.CurrentX = m / 10
PictureBernoulli.CurrentY = 1.1
PictureBernoulli.Print "Px"
PictureBernoulli.CurrentX = n + m / 2
PictureBernoulli.CurrentY = 0.05
PictureBernoulli.Print "X"
'Numerazione
PictureBernoulli.CurrentX = -m / 2
PictureBernoulli.CurrentY = -0.02
PictureBernoulli.Print "o"
x = CInt(m)
If x = 0 Then x = 1
For i = 1 To n Step x
    PictureBernoulli.CurrentX = i - PicturePoisson.TextWidth(CStr(i)) / 2
    PictureBernoulli.CurrentY = -0.02
    PictureBernoulli.Print CStr(i)
Next i
For i = 0.1 To 1 Step 0.1
    PictureBernoulli.CurrentX = -m / 1.2
    PictureBernoulli.CurrentY = i + 0.02
    PictureBernoulli.Print CStr(i)
    PictureBernoulli.Line (-m / 10, i)-(m / 10, i), RGB(0, 0, 255)
Next i
'Stampa Bernoulli
PictureBernoulli.FontSize = 36
PictureBernoulli.FontName = "Times New Roman"
PictureBernoulli.ForeColor = RGB(180, 180, 180)
PictureBernoulli.CurrentX = n / 2 - PictureBernoulli.TextWidth("Bernoulli") / 2
PictureBernoulli.CurrentY = 0.7
PictureBernoulli.Print "Bernoulli"
PictureBernoulli.ForeColor = vbBlack
PictureBernoulli.FontSize = 14
PictureBernoulli.FontName = "MS Sans Serif"
PictureBernoulli.CurrentX = n / 2 - PictureBernoulli.TextWidth("p = " & CStr(p) _
    & " q = " & CStr(1 - p)) / 2
PictureBernoulli.CurrentY = 1.1
PictureBernoulli.Print "p = " & CStr(p) & " q = " & CStr(1 - p)
PictureBernoulli.FontSize = 8
If Zero = True And n = 1 Then n = 0: Zero = False
End Sub
```

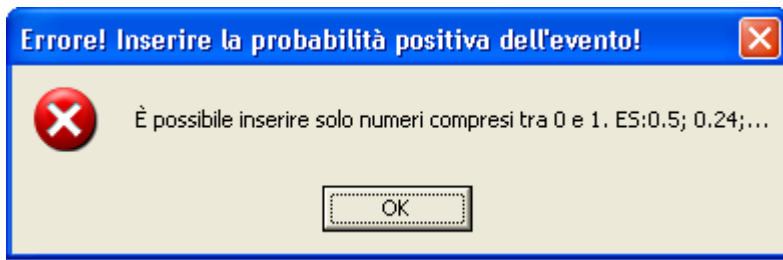
---

```
Public Sub DisegnaBernoulli(j As Integer)
PictureBernoulli.DrawWidth = 3
PictureBernoulli.Line (j, 0)-(j, TabellaB(j)), RGB(0, 255, 0)
PictureBernoulli.DrawWidth = 1
End Sub
```

### **Interfaccia:**



Se l'utente immette valori errati sarà avvertito con questi messaggi, rispettivamente quello per l'inserimento di n e quello di p.



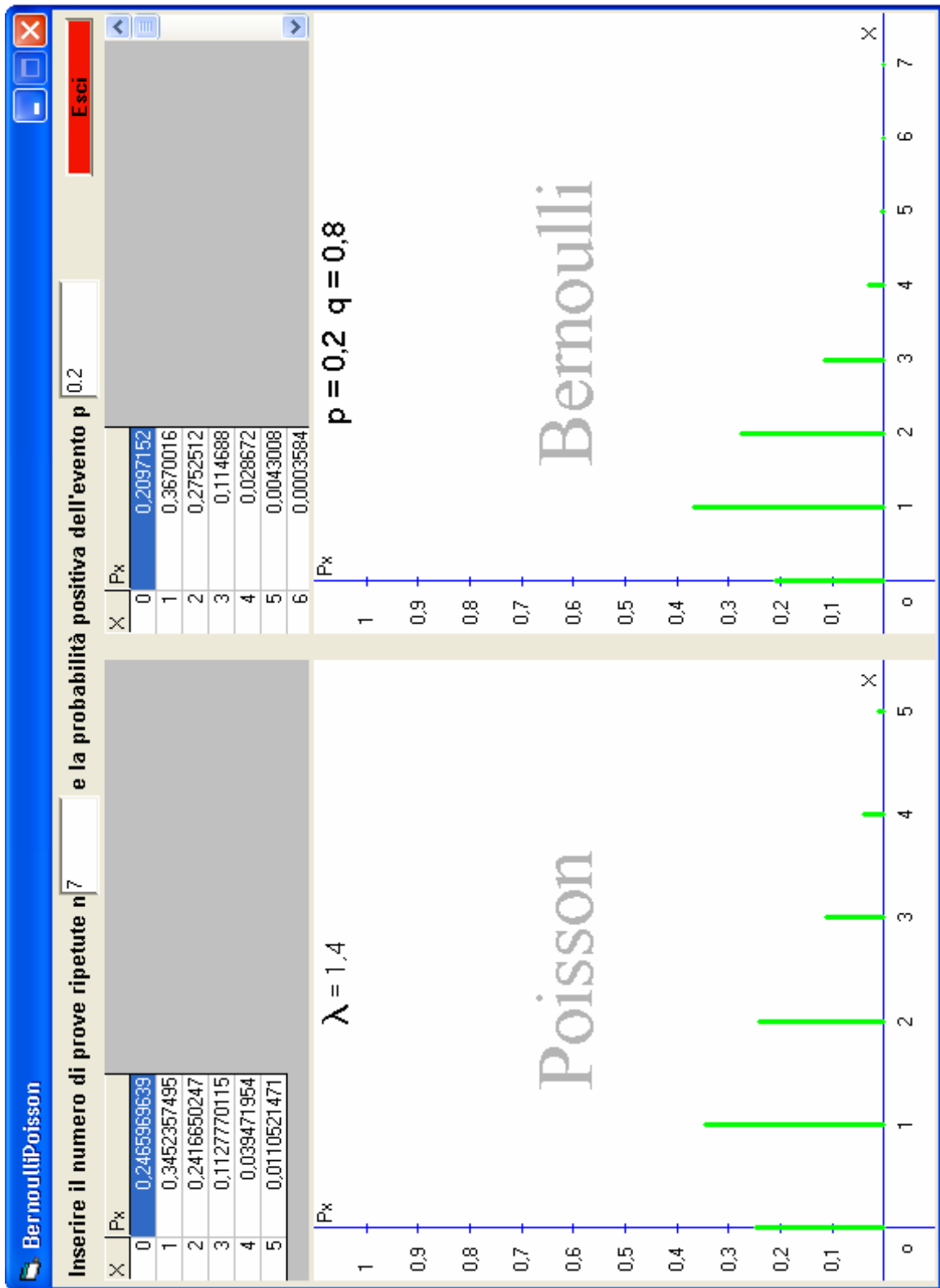
Questo è il Form in fase di progettazione, le due tabelle e le due griglie hanno la stessa dimensione e sono posti parallelamente per facilitare il confronto.



**Verificare che, al crescere di n, le distribuzioni tendono a sovrapporsi:**

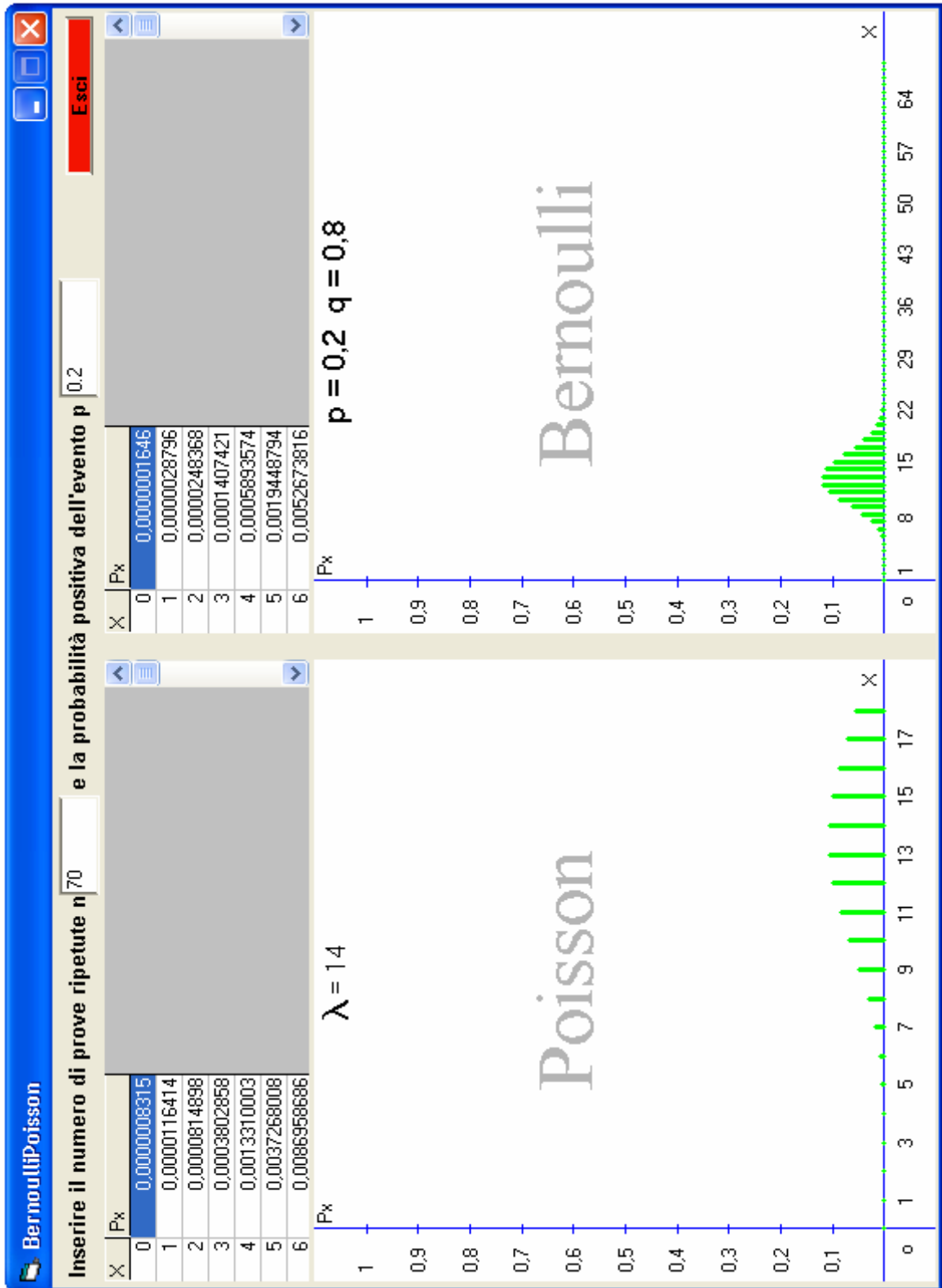
Nel prossimo Form mostro un esempio di come si presentano le output se l'utente digita 7 come numero di determinazioni e 0.2 come probabilità positiva.

È da notare come sono simili i valori, come si vede dalle tabelle, utilizzando entrambi i metodi.



Ponendo  $n = 70$  è evidente che gli effetti si sovrappongono, soprattutto in Bernoulli dove sono visualizzati tutti i  $n$  valori.

Per non far sovrapporre la numerazione dell'asse x vengono stampati solo 10 numeri. Nei grafici sono presenti anche il valore di Lambda, la probabilità positiva ( $p$ ) e quella negativa ( $q$ ). Per facilitare l'utente a comprendere il risultato.



**Ciao Ciao!**

Grazie per aver scelto questo programma.  
Arrivederci alla prossima.

OK

Firma \_\_\_\_\_