

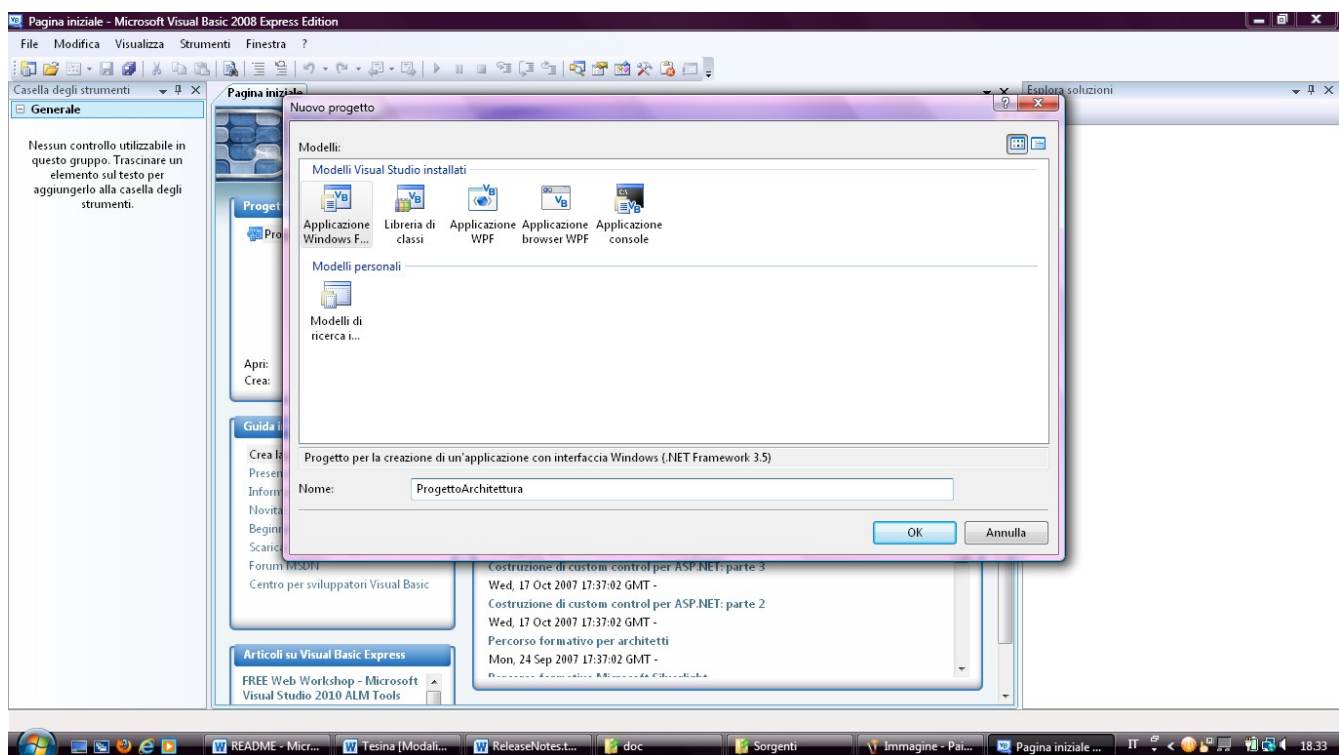
Note di rilascio del simulatore Ordinamento dei Byte , versione 1.0

Architettura del software

Per la costruzione dell'applicazione software, è stato utilizzato l'ambiente di programmazione visuale **Visual Basic** versione **2008**, che permetterà attraverso gli **oggetti** della programmazione, ovvero **form** e **controlli**, la realizzazione di un'interfaccia grafica per gestire la nostra base di dati.

Si vedano adesso i vari passi per la creazione del programma finale per l'utente.

Per prima cosa è importante creare un nuovo progetto, ovvero una nuova **Applicazione Windows Form** che prenderà il nome di *ProgettoArchitettura*.

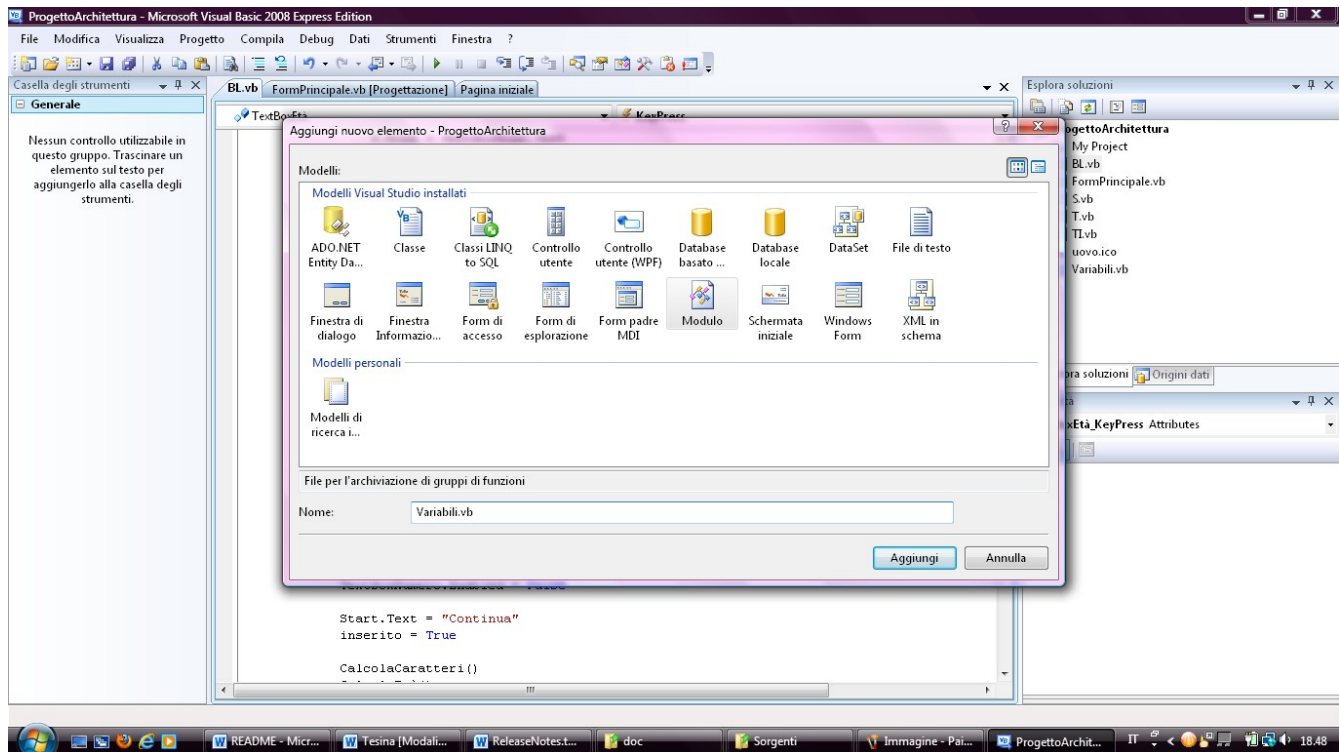


A questo punto nella parte della finestra di programmazione, riguardante la gestione del progetto (ovvero *Esplora soluzioni*), troveremo già due elementi di default:

- My Project, per impostare le caratteristiche del progetto;
- *Form1.vb*, che costituirà il Form principale di tutta l'applicazione

Modulo : *Variabili.vb*

Una parte importante dell'applicazione software è quella che riguarda la dichiarazione delle variabili contenuta in un **modulo**. Per aggiungere un modulo all'interno del programma, basta selezionare l'opzione “*Aggiungi elemento*” dalla barra del Menù, e procedere nel seguente modo:



Il nuovo modulo prende il nome di *Variabili.vb*.

Poi si passa alla dichiarazione della **struttura delle variabili**: nel programma viene utilizzato una singola variabile di tipo *Record*, dopo averne definito la struttura.

La variabile principale è chiamata *R*: in tale variabile verranno memorizzati i dati inseriti dall'utente. La struttura, che prende il nome di *Record*, del tipo della variabile è costituita da 4 campi:

- Un campo *Nome* di tipo *String* contenente il nome dell'impiegato;
- Un campo *Cognome* di tipo *String* contenente il cognome dell'impiegato;
- Un campo *Età* di tipo *Integer* contenente l'età dell'impiegato;
- Un campo *Numero* di tipo *Integer* contenente il numero del reparto dell'impiegato;

Module Variabili

```
'Struttura del Record
Public Structure Record
    Dim Nome As String
    Dim Cognome As String
    Dim Et  As Integer
    Dim Numero As Integer
End Structure

'Variabile di tipo Record
Public R As Record
```

Dopo aver dichiarato la struttura del tipo della variabile principale e la variabile principale stessa, vengono dichiarate le **variabili globali**, aventi ciascuna un ruolo ben preciso:

- L'Array *C* di tipo *Char* servir  per memorizzare il Nome e il Cognome dell'impiegato separati da uno spazio. Se l'intero Array non risulter  pieno, verr  riempito da 0;
- L'Array *E* di tipo *Integer* servir  per memorizzare l'et  dell'impiegato;
- L'Array *N* di tipo *Integer* servir  per memorizzare il numero del reparto dell'impiegato;
- La variabile *num* di tipo *Integer*, inizializzata a -1, servir  per il **Windows Form** *BL.vb* e verr  illustrata successivamente;

```
'Array di caratteri
Public C(0 To 11) As Char
'Array di interi
Public E(0 To 3) As Integer
Public N(0 To 3) As Integer

'Variabile intera globale che svolge il ruolo di contatore utilizzata nei Form
'"FormPrincipale" e "BL"
Public num As Integer = -1
```

Windows Form : FormPrincipale.vb

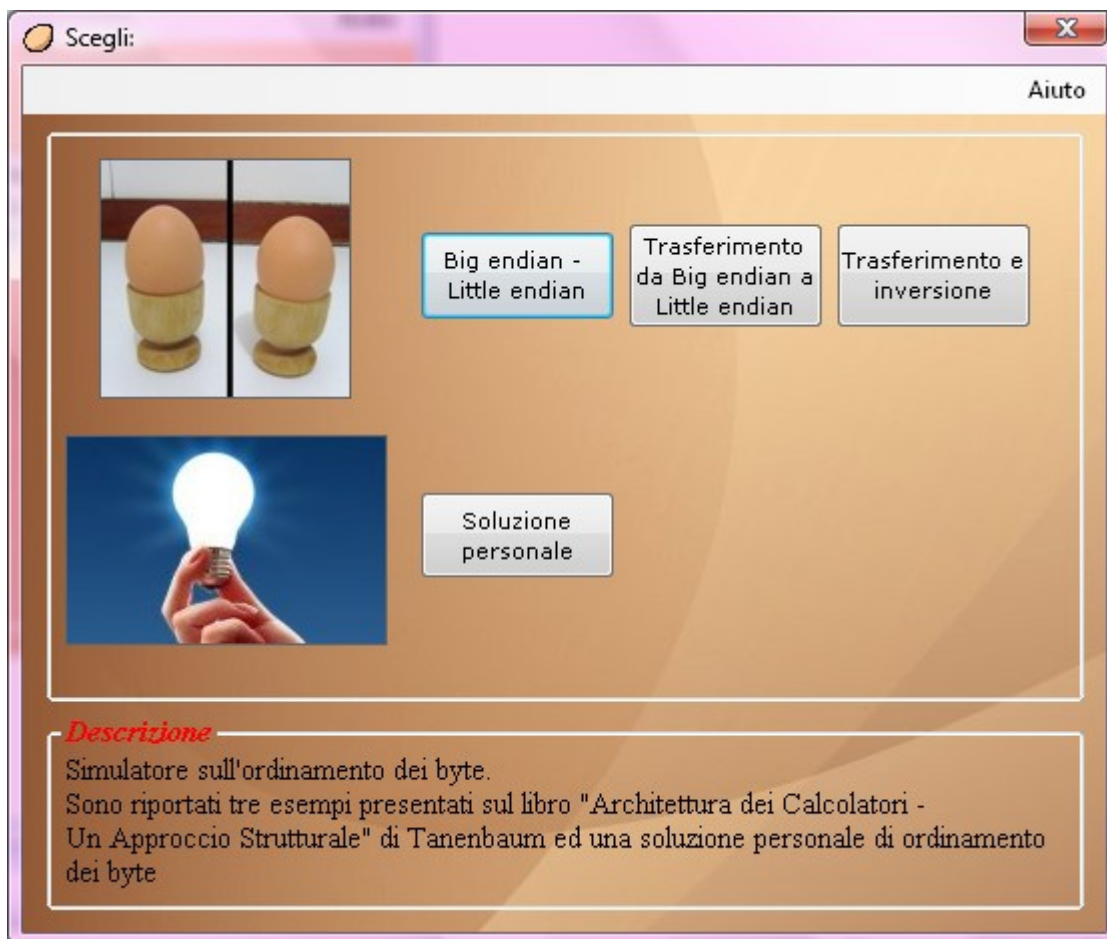
Nel **Windows Form** *FormPrincipale.vb* vengono impostate nella parte della finestra riguardanti la gestione delle propriet  degli oggetti (ovvero *Propriet *):

- La dimensione attraverso la propriet  *Size*, con *Width* 557 e *Height* 468, in pixel;
- Lo sfondo attraverso la propriet  *BackgroundImage*, scegliendo un'immagine di sfondo di formato *.jpg* a piacere;
- L'intestazione della form attraverso la propriet  *Text* con "Scegli:";
- L'impossibilit  di poter ingrandire il Form1 con l'utilizzo del pulsante d'ingrandimento posto nella barra del Men  nell'angolo in alto a destra attraverso la propriet  *MaximizeBox* uguale a *False*;
- L'icona della form attraverso la propriet  *Icon*, scegliendo un'icona di formato *.ico* a piacere;

Dopo aver impostate le caratteristiche del **Windows Form** *FormPrincipale.vb*, vengono introdotti al suo interno alcuni controlli, quali:

- Quattro **Windows Form Button**, a cui saranno associate delle funzioni da fare in seguito all'evento *Click* di uno specifico pulsante. I quattro oggetti **Windows Form Button** sono posti nel seguente ordine e prendono il nome di:
 - **BigLittle**, al cui click comparirà un nuovo **Windows Form**, di nome "*BL.vb*", per far inserire dei dati all'utente e successivamente visualizzare il modo in cui tali dati sono memorizzati all'interno di una macchina *Big-endian* e una macchina *Little-endian*; la proprietà *Text* è stata impostata a "*Big endian – Little endian*".
 - **Trasf**, al cui click comparirà un nuovo **Windows Form**, di nome "*T.vb*", per far inserire dei dati all'utente e successivamente visualizzare il modo in cui tali dati vengono memorizzati all'interno di una macchina *Big-endian* e trasferiti in una macchina *Little-endian*; la proprietà *Text* è stata impostata a "*Trasferimento da Big endian a Little endian*".
 - **TraInv**, al cui click comparirà un nuovo **Windows Form**, di nome "*TI.vb*", per far inserire dei dati all'utente e successivamente visualizzare il modo in cui tali dati vengono memorizzati all'interno di una macchina *Little-endian* dopo il processo di trasferimento da una macchina *Big-endian* e il modo in cui tali dati vengono memorizzati sempre all'interno di una macchina *Little-endian* dopo il processo di trasferimento e inversione da una macchina *Big-endian*; la proprietà *Text* è stata impostata a "*Trasferimento e Inversione*".
 - **Sol**, al cui click comparirà un nuovo **Windows Form**, di nome "*S.vb*", per far inserire dei dati all'utente e successivamente visualizzare il modo in cui tali dati vengono memorizzati all'interno di una macchina *Big-endian* e trasferiti ad una macchina *Little-endian*, eliminando i problemi sull'ordine dei byte che si creano in seguito al trasferimento; la proprietà *Text* è stata impostata a "*Soluzione personale*".
- Due **Windows Form PictureBox**, di dimensioni diverse, al cui interno sono state inserite due immagini di formato *.jpg* scelte per rendere meglio l'idea delle funzionalità del programma;
- Un **Windows Form Label**, in cui è presente una breve descrizione testuale delle funzionalità che il programma è in grado di eseguire;
- Due **Windows Form GroupBox**, di dimensioni diverse, per visualizzare una cornice contenente i vari controlli che possono essere inseriti in un **Windows Form**; nel nostro caso uno conterrà i quattro **Windows Form Button** e i due **Windows Form PictureBox** mentre l'altro conterrà il **Windows Form Label**;

In questo modo la schermata principale, o il **Windows Form** principale, di tutto il programma presenta il seguente aspetto:



Windows Form Button: *BigLittle*

Una delle prime funzioni che si consiglia di far eseguire al programma non appena viene avviato, è quella di visualizzare il modo in cui un dato, che sia in forma numerica o letterale, purché inserito dall'utente, viene memorizzato all'interno di una macchina *Big-endian* e *Little-endian*, cioè in particolare il modo in cui vengono disposti i relativi byte del dato inserito.

Facendo un esempio (*Esempio 1.0*):

Consideriamo un semplice archivio per il personale con record costituiti da una stringa (il nome e il cognome dell'impiegato separati da uno spazio) e due interi (l'età e il numero del reparto). La stringa termina con uno o più byte con valore 0 in modo da riempire interamente una parola. La **Figura 1.1** mostra la rappresentazione *Big-endian* di *Jim Smith*, età 21 anni, dipartimento 260 ($1 \times 256 + 4 = 260$); la rappresentazione *Little-endian* è data invece nella **Figura 1.2**. Le due rappresentazioni sono corrette e coerenti internamente.

Big-endian

Indirizzo

Little-endian

	M	I	J
4	T	I	S
8	0	0	0
12	0	0	0
16	0	0	1
			4

Figura 1.1

Figura 1.2

E' importante capire che sia nei sistemi *Big-endian* sia in quelli *Little-endian* un intero a 32 bit, per esempio con valore numerico 6, è rappresentato dai bit *110* nei 3 bit più a destra (meno significativi) di una parola e *zero* nei 29 bit più a sinistra. Nello schema di tipo *Big-endian* i bit *110* si trovano nel byte 3 (o 7, o 11, ecc.), mentre in quello di tipo *Little-endian* nel byte 0 (o 4, o 8, ecc.). In entrambi i casi la parola che contiene questo indirizzo ha indirizzo 0 (o 4, o 8, ecc.).

Questo è il compito del **Windows Form Button *BigLittle***: al suo evento *Click*, viene associato il seguente codice:

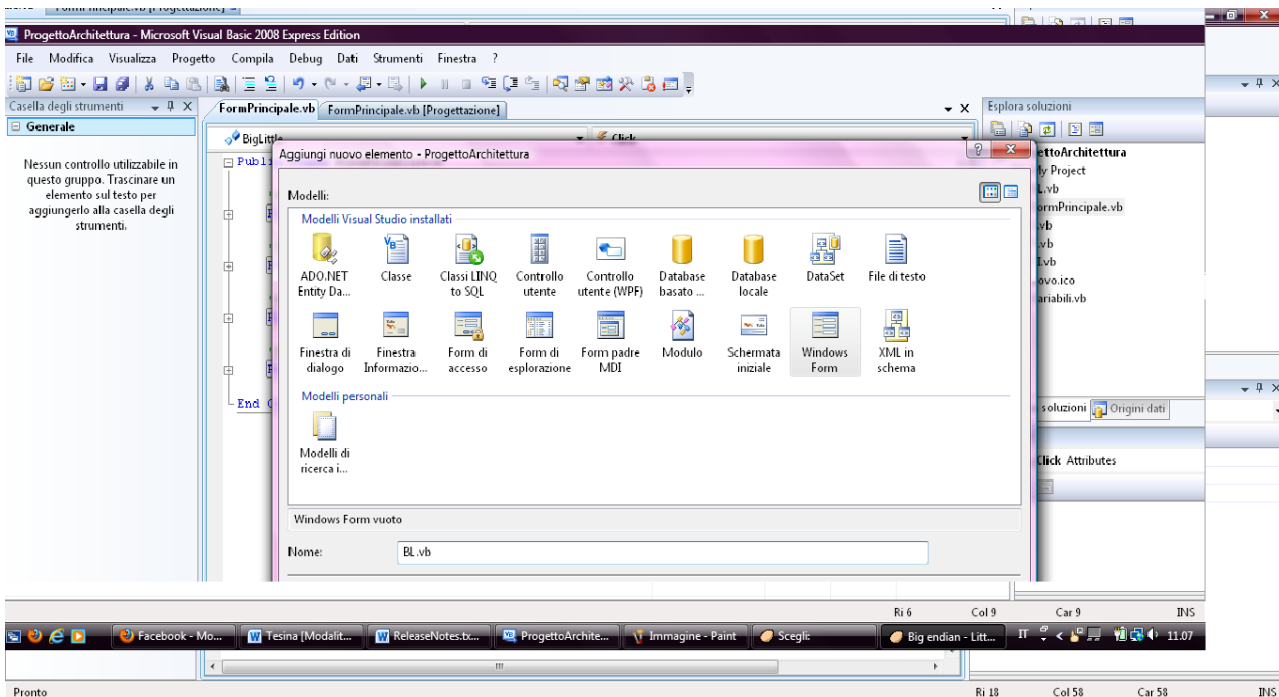
```
'Funzione associata all'evento click del button BigLittle
Private Sub BigLittle_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles BigLittle.Click
    'Variabile intera globale che svolge il ruolo di contatore per la visualizzazione
    num = -1
    'Viene mostrato il form BL
    BL.Show()
End Sub
```

Viene impostato il valore della variabile intera *num* a *-1*, che svolge il ruolo di contatore e che verrà richiamata successivamente all'interno del **Windows Form *BL.vb***. E' quindi possibile trovare maggiori informazioni all'interno della spiegazione del **Windows Form *BL.vb***.

Dopodiché si passa alla visualizzazione del **Windows Form *BL.vb***, attraverso il richiamo della funzione *BL.Show()*.

Windows Form : *BL.vb*

Il seguente **Windows Form *BL.vb*** permette di inserire all'utente dei dati arbitrari e di visualizzare il modo in cui i singoli byte di tali dati vengono disposti all'interno di una macchina *Big-endian* e *Little-endian*. Per aggiungere un nuovo form nel progetto basta selezionare l'opzione "*Aggiungi elemento*" della barra del Menù e procedere nel seguente modo:



Nel **Windows Form** *BL.vb* vengono impostate nella parte della finestra riguardanti la gestione delle proprietà degli oggetti (ovvero *Proprietà*):

- La dimensione attraverso la proprietà *Size*, con *Width* 784 e *Height* 553, in pixel;
- Lo sfondo attraverso la proprietà *BackgroundImage*, scegliendo un'immagine di sfondo di formato *.jpg* a piacere;
- L'intestazione della form attraverso la proprietà *Text* con "*Big endian – Little endian*";
- L'impossibilità di poter ingrandire il **Windows Form** *BL.vb* con l'utilizzo del pulsante d'ingrandimento posto nella barra del Menù nell'angolo in alto a destra attraverso la proprietà *MaximizeBox* uguale a *False*;
- L'icona della form attraverso la proprietà *Icon*, scegliendo un'icona di formato *.ico* a piacere;

Dopo aver impostato le caratteristiche del **Windows Form** *BL.vb*, vengono introdotti al suo interno alcuni controlli, quali:

- Quattro **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, a cui è stato impostato:
 - Lo sfondo attraverso la proprietà *BackColor* uguale a *Transparent* per rendere lo sfondo dell'oggetto **Windows Form Label** trasparente;
 - L'intestazione attraverso la proprietà *Text* con "*Nome*" per il **Windows Form Label** *LabelNome*, "*Cognome*" per il **Windows Form Label** *LabelCognome*, "*Età*" per il **Windows Form Label** *LabelEtà* e "*Numero reparto*" per il **Windows Form Label** *LabelNumero*;
- Venti **Windows Form Label** (*LabelL0*, *LabelL1*, *LabelL2*, ..., *LabelL19*) e venti **Windows Form Label** (*LabelB0*, *LabelB1*, *LabelB2*, ..., *LabelB19*) all'interno di

due **Table Layout Panel**, che visualizzeranno entrambi una numerazione da 0 a 19 inclusi;

- Due **Table Layout Panel**, in cui sarà visualizzata la disposizione dei byte dei dati inseriti dall'utente, ovviamente diversa per ciascun **Table Layout Panel** in quanto uno si riferisce ad una macchina *Big-endian* e l'altro ad una macchina *Little-endian*;
- Quattro **Windows Form TextBox** (*TextBoxNome*, *TextBoxCognome*, *TextBoxEtà* e *TextBoxNumero*) posizionati uno sotto l'altro accanto, rispettivamente, al **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, per acquisire dall'utente i dati da inserire nella macchina *Big-endian* e *Little-endian*;
- Un **Windows Form Button**, la cui proprietà *Text* è inizialmente "Start!"; in seguito al primo evento *Click* verrà cambiata la proprietà *Text* in "Continua", i quattro **Windows Form TextBox** verranno disabilitati e i byte dei dati inseriti dall'utente verranno disposti in maniera diversa all'interno dei due **Table Layout Panel**, per rendere visibile all'utente il meccanismo differente in cui i singoli byte vengono memorizzati all'interno di una macchina *Big-endian* e *Little-endian*; al 22° evento *Click* la proprietà *Text* sarà impostata a "Start!" per permettere all'utente di ricominciare nuovamente il procedimento con dati differenti;
- Un **Windows Form GroupBox**, la cui proprietà *Text* è impostata a "Descrizione" e la proprietà *BackColor* a "Trasparent", che svolge il ruolo di cornice per contenere al suo interno un **Windows Form Label**, che permetterà all'utente di essere informato attraverso una breve descrizione della specifica sezione del programma che è stata avviata;

In questo modo il **Windows Form** appena creato dell'applicazione presenta il seguente aspetto:

Big endian - Little endian

	Big-endian				Indirizzo	Little-endian			
Nome	0	1	2	3	0	3	2	1	0
Cognome	4	5	6	7	4	7	6	5	4
Età	8	9	10	11	8	11	10	9	8
Numero reparto	12	13	14	15	12	15	14	13	12
	16	17	18	19	16	19	18	17	16

Parola a 32 bit

Descrizione

Il sistema a sinistra, in cui la numerazione comincia a partire dall'estremo più "grande" (cioè dal byte più significativo) è chiamato big endian, in contrapposizione con il sistema a destra little endian, in cui la numerazione comincia a partire dall'estremo più "piccolo" (cioè dal byte meno significativo). E' importante capire che sia nei sistemi big endian sia in quelli little endian un intero a 32 bit, per esempio con valore numerico 6, è rappresentato dai bit 110 nei 3 bit più a destra (meno significativi) di una parola e zero nei 29 bit più a sinistra. Nello schema di tipo big endian i bit 110 si trovano nel byte 3 (o 7, o 11, ecc.), mentre in quello di tipo little endian nel byte 0 (o 4, o 8, ecc.). In entrambi i casi la parola che contiene questo indirizzo ha indirizzo 0.

Se i calcolatori memorizzassero soltanto gli interi non ci sarebbe nessun problema. Tuttavia molte applicazioni richiedono interi, stringhe di caratteri e altri tipi di dati. In questa sezione del software viene considerato per esempio un semplice archivio per il personale con record costituiti da due stringhe (il nome e il cognome dell'impiegato) e due interi (l'età e il numero del reparto). L'utente potrà quindi inserire i valori da lui scelti e vedere come tali valori verranno memorizzati in un sistema big endian e little endian. La stringa (composta da nome e cognome) terminerà con uno o più byte con valore 0 in modo da riempire interamente una parola.

Dopo aver creato un nuovo **Windows Form**, è importante soprattutto gestire le risposte dell'applicazione in seguito agli eventi dell'utente o del sistema.

Il primo codice da inserire è quello che riguarda l'evento *Load*, in cui viene impostata la proprietà *Text* dei quattro **Windows Form TextBox** a “*Nothing*” e richiamata la funzione *Numera_Label* che inizierà il valore dei 40 **Windows Form Label** per motivi grafici.

```
'Funzione associata all'apertura del form BL
Private Sub BL_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles MyBase.Load
    TextBoxNome.Text = Nothing
    TextBoxCognome.Text = Nothing
    TextBoxEtà.Text = Nothing
    TextBoxNumero.Text = Nothing

    'Richiamo della funzione che inizializza la visualizzazione dei 40 Label all'interno
    'delle due tabelle
    Numera_Label()
End Sub
```

La funzione *Numera_Label* cambia la proprietà *Text* di ogni singolo **Windows Form Label** assegnandogli un valore numerico. Ciò servirà per motivi grafici, ovvero all'avvio del **Windows Form BL.vb** apparirà una numerazione delle caselle delle due **Tablet Layout Panel** per spiegare meglio all'utente l'ordinamento dei byte nel processo di memorizzazione di un dato (numerico o letterale) all'interno di una macchina *Big-endian* o *Little-endian*.

```

'Funzione che inizializza la visualizzazione dei 40 Label all'interno delle due tabelle
Private Sub Numera_Label()
    LabelB0.Text = "0"
    LabelB1.Text = "1"
    LabelB2.Text = "2"
    LabelB3.Text = "3"
    LabelB4.Text = "4"
    LabelB5.Text = "5"
    LabelB6.Text = "6"
    LabelB7.Text = "7"
    LabelB8.Text = "8"
    LabelB9.Text = "9"
    LabelB10.Text = "10"
    LabelB11.Text = "11"
    LabelB12.Text = "12"
    LabelB13.Text = "13"
    LabelB14.Text = "14"
    LabelB15.Text = "15"
    LabelB16.Text = "16"
    LabelB17.Text = "17"
    LabelB18.Text = "18"
    LabelB19.Text = "19"

    LabelL0.Text = "0"
    LabelL1.Text = "1"
    LabelL2.Text = "2"
    LabelL3.Text = "3"
    LabelL4.Text = "4"
    LabelL5.Text = "5"
    LabelL6.Text = "6"
    LabelL7.Text = "7"
    LabelL8.Text = "8"
    LabelL9.Text = "9"
    LabelL10.Text = "10"
    LabelL11.Text = "11"
    LabelL12.Text = "12"
    LabelL13.Text = "13"
    LabelL14.Text = "14"
    LabelL15.Text = "15"
    LabelL16.Text = "16"
    LabelL17.Text = "17"
    LabelL18.Text = "18"
    LabelL19.Text = "19"

End Sub

```

A questo punto bisogna gestire cosa dovrà fare l'applicazione in risposta all'inserimento di un valore all'interno di una **Windows Form TextBox**.

Nel caso di inserimento di un valore all'interno della **Windows Form TextBox** *TextBoxNome* o *TextBoxCognome*, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale.

Quando l'utente digita un tasto da tastiera, partono una serie di controlli affinché il tasto digitato sia esclusivamente alfabetico, consentendo anche all'utente di poter digitare il tasto *BackSpace* ed il tasto *Canc*. Tali controlli vengono effettuati analizzando il corrispettivo codice ASCII.

Se il tasto digitato non rientra nei canoni sopra citati, questo non verrà accettato dal programma e quindi non verrà visualizzato all'interno della **Windows Form TextBox**.

Infine è bene tenere presente che è consentito solo inserire non più di 5 caratteri all'interno della **Windows Form TextBox** *TextBoxNome* o *TextBoxCognome*: il motivo è spiegato nell'ultima sezione del documento chiamata "Idee".

Windows Form TextBox *TextBoxNome*

```

'Funzione che controlla che l'inserimento da parte dell'utente all'interno della TextBox
'"TextBoxNome" sia solo di tipo alfabetico
Private Sub TextBoxNome_KeyPress(ByVal sender As Object, ByVal e _
    As System.Windows.Forms.KeyPressEventArgs) Handles TextBoxNome.KeyPress

    If (TextBoxNome.TextLength < 5) Then

        ' Accetto solo l'immissione di lettere
        Dim KeyAscii As Short = Asc(e.KeyChar)

        ' Le lettere maiuscole hanno i codici ascii tra
        ' 65 e 90, quelle minuscole tra 97 e 122.
        ' Devo comunque fare in modo che l'utente
        ' sia in grado di digitare anche il tasto BackSpace
        ' (ascii=8) e il tasto Canc (ascii=24)
        If KeyAscii < 65 And KeyAscii <> 24 And KeyAscii <> 8 Then
            KeyAscii = 0
        ElseIf KeyAscii > 90 And KeyAscii < 97 Then
            KeyAscii = 0
        ElseIf KeyAscii > 122 Then
            KeyAscii = 0
        End If

        ' Reimposto il KeyChar
        e.KeyChar = Chr(KeyAscii)
    Else
        e.KeyChar = Nothing
    End If

End Sub

```

Windows Form TextBox *TextBoxCognome*

```

'Funzione che controlla che l'inserimento da parte dell'utente all'interno della TextBox
'"TextBoxCognome" sia solo di tipo alfabetico
Private Sub TextBoxCognome_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles TextBoxCognome.KeyPress

    If (TextBoxCognome.TextLength < 5) Then
        ' Accetto solo l'immissione di lettere

        Dim KeyAscii As Short = Asc(e.KeyChar)

        ' Le lettere maiuscole hanno i codici ascii tra
        ' 65 e 90, quelle minuscole tra 97 e 122.
        ' Devo comunque fare in modo che l'utente
        ' sia in grado di digitare anche il tasto BackSpace
        ' (ascii=8) e il tasto Canc (ascii=24)
        If KeyAscii < 65 And KeyAscii <> 24 And KeyAscii <> 8 Then
            KeyAscii = 0
        ElseIf KeyAscii > 90 And KeyAscii < 97 Then
            KeyAscii = 0
        ElseIf KeyAscii > 122 Then
            KeyAscii = 0
        End If

        ' Reimposto il KeyChar
        e.KeyChar = Chr(KeyAscii)
    Else
        e.KeyChar = Nothing
    End If

End Sub

```

Non tanto diversa è la situazione della **Windows Form TextBox** *TextBoxEtà* o *TextBoxNumero*: nel caso in cui viene inserito un valore, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale.

Quando l'utente digita un tasto da tastiera, partono una serie di controlli affinché il tasto digitato sia esclusivamente numerico, consentendo anche all'utente di poter digitare il tasto *BackSpace* ed il tasto *Canc*. Tali controlli vengono effettuati analizzando il corrispettivo codice ASCII.

Se il tasto digitato non rientra nei canoni sopra citati, questo non verrà accettato dal programma e quindi non verrà visualizzato all'interno della **Windows Form TextBox**.

Infine è bene tenere presente che non è stato gestito il controllo sulla lunghezza del dato inserito all'interno della **Windows Form TextBox** *TextBoxEtà* o *TextBoxNumero*: sarebbe bene tenere in considerazione questa modifica che verrà meglio spiegata nell'ultima sezione del documento chiamata "*Idee*".

Windows Form TextBox *TextBoxEtà*

```

'Funzione che controlla che l'inserimento da parte dell'utente all'interno della TextBox
'"TextBoxEtà" sia solo di tipo numerico
Private Sub TextBoxEtà_KeyPress(ByVal sender As Object, ByVal e _
    As System.Windows.Forms.KeyPressEventArgs) Handles TextBoxEtà.KeyPress

    ' Accetto solo l'immissione di numeri interi

    ' Recupero il codice ascii del tasto digitato
    ' il tasto digitato è memorizzato nella proprietà "KeyChar"
    ' dell'oggetto System.Windows.Forms.KeyPressEventArgs
    Dim KeyAscii As Short = Asc(e.KeyChar)

    ' I numeri interi hanno il codice ascii compreso tra
    ' 48 e 57. Devo comunque fare in modo che l'utente
    ' sia in grado di digitare anche il tasto BackSpace
    ' (ascii=8) e il tasto Canc (ascii=24), se il codice
    ' ascii non rientra in quelli ammessi, lo imposto io
    ' su Zero, che è il carattere nullo.
    If KeyAscii < 48 And KeyAscii <> 24 And KeyAscii <> 8 Then
        KeyAscii = 0
    ElseIf KeyAscii > 57 Then
        KeyAscii = 0
    End If

    ' Aggiungo un'ulteriore finezza facendo in modo che lo
    ' zero sia ammesso, ma non come primo carattere, lo faccio
    ' controllando la lunghezza del testo
    If e.KeyChar = "0" And TextBoxEtà.TextLength = 0 Then
        KeyAscii = 0
    End If

    ' Alla fine reimposto il KeyChar, che sarà uguale a zero
    ' per i caratteri non consentiti (e quindi nella TextBox
    ' non comparirà nulla)
    e.KeyChar = Chr(KeyAscii)

End Sub

```



```

'Funzione che controlla che l'inserimento da parte dell'utente all'interno della TextBox
'"TextBoxNumero" sia solo di tipo numerico
Private Sub TextBoxNumero_KeyPress(ByVal sender As Object, ByVal e _
    As System.Windows.Forms.KeyPressEventArgs) Handles TextBoxNumero.KeyPress

    ' Accetto solo l'immissione di numeri interi

    ' Recupero il codice ascii del tasto digitato
    ' il tasto digitato è memorizzato nella proprietà "KeyChar"
    ' dell'oggetto System.Windows.Forms.KeyPressEventArgs
    Dim KeyAscii As Short = Asc(e.KeyChar)

    ' I numeri interi hanno il codice ascii compreso tra
    ' 48 e 57. Devo comunque fare in modo che l'utente
    ' sia in grado di digitare anche il tasto BackSpace
    ' (ascii=8) e il tasto Canc (ascii=24), se il codice
    ' ascii non rientra in quelli ammessi, lo imposto io
    ' su Zero, che è il carattere nullo.
    If KeyAscii < 48 And KeyAscii <> 24 And KeyAscii <> 8 Then
        KeyAscii = 0
    ElseIf KeyAscii > 57 Then
        KeyAscii = 0
    End If

    ' Aggiungo un'ulteriore finezza facendo in modo che lo
    ' zero sia ammesso, ma non come primo carattere, lo faccio
    ' controllando la lunghezza del testo
    If e.KeyChar = "0" And TextBoxNumero.TextLength = 0 Then
        KeyAscii = 0
    End If

    ' Alla fine reimposto il KeyChar, che sarà uguale a zero
    ' per i caratteri non consentiti (e quindi nella TextBox
    ' non comparirà nulla)
    e.KeyChar = Chr(KeyAscii)

End Sub

```

Adesso possiamo passare ad esaminare ciò che succede quando viene premuto il pulsante “*Start!*”, ovvero iniziamo a capire il codice associato all’evento *Click* del **Windows Form Button Start**.

Al click del **Windows Form Button Start** si eseguono le seguenti operazioni: per prima cosa bisogna assicurarsi che ogni **Windows Form TextBox** non sia vuota, cioè che la proprietà *Text* sia diversa da “*Nothing*”: in questo caso verrà mandato un **MessageBox** d’errore all’utente, altrimenti ogni valore della proprietà *Text* verrà copiato in un corrispondente campo della Struttura Record *R*.

Successivamente le quattro **Windows Form TextBox** vengono disabilitate impostando la loro proprietà *Enabled* a “*False*”, la proprietà *Text* del **Windows Form Button Start** viene impostata a “*Continua*” e il valore della variabile booleana *inserito* viene impostato a “*True*”, così, ad un secondo click del **Windows Form Button Start**, non si entrerà più nella parte di codice riguardante i controlli dei valori delle quattro **Windows Form TextBox**.

La fase successiva è l’esecuzione di tre funzioni : *CalcolaCaratteri*, *CalcolaEtà*, *CalcolaNumero*. In queste tre funzioni grossolanamente vengono passati i valori delle quattro **Windows Form TextBox** in 3 diversi **Array**, in particolare:

- nella funzione *CalcolaCaratteri*, vengono copiati i singoli caratteri inseriti dall'utente all'interno della **Windows Form TextBox** *TextBoxNome* all'interno dell'**Array** *C* a partire dalla posizione 0 e poi vengono copiati i singoli caratteri inseriti dall'utente all'interno della **Windows Form TextBox** *TextBoxCognome* sempre all'interno dell'**Array** *C* a partire dalla posizione $n+1$, dove n è la lunghezza della stringa inserita all'interno della **Windows Form TextBox** *TextBoxNome* (il “+1” stà ad indicare uno spazio tra il nome ed il cognome all'interno dell'**Array** *C*);
- nelle funzioni *CalcolaEtà* e *CalcolaNumero*, vengono trasformati i valori di tipo stringa, inseriti dall'utente all'interno della **Windows Form TextBox** *TextBoxEtà* e *TextBoxNumero*, in valori di tipo numerico e memorizzati ciascuno all'interno di una variabile d'appoggio. In seguito ogni numero è convertito in valore binario per essere poi diviso in quattro gruppi da 8 bit: di ogni gruppo si calcola il corrispettivo valore decimale, ottenendo così quattro numeri decimali, e lo si memorizza all'interno dell'**Array** *E*, nel caso della funzione *CalcolaEtà*, o dell'**Array** *N*, nel caso della funzione *CalcolaNumero*;

A questo punto l'utente sarà obbligato a richiamare l'evento *Click* del **Windows Form Button** *Start* per almeno 21 volte, prima di poter riabilitare le quattro **Windows Form TextBox** e poter reinserire nuovi dati.

Richiamando quindi l'evento *Click* del **Windows Form Button** *Start* si passa così all'incremento della variabile *num* e all'esecuzione della funzione *Controlla* alla quale viene passata come parametro proprio il valore della variabile *num*.

Quando la variabile *num* raggiungerà il valore 20, allora il valore della variabile booleana *inserito* viene impostata a “False”, la proprietà *Text* del **Windows Form Button** *Start* è uguale a “Start!”, il valore della variabile *num* è uguale a “-1”, le quattro **Windows Form TextBox** vengono riabilite, impostando la proprietà *Enabled* a “True”, e infine viene richiamata la funzione *Numera_Label* già vista in precedenza.

In seguito vi è una spiegazione più approfondita delle funzioni *CalcolaCaratteri*, *CalcolaEtà*, *CalcolaNumero* e *Controlla*.

```

'Funzione associata all'evento click del Button "Start".
'Se viene cliccato il Button "Start" ed una delle quattro TextBox è vuota, allora viene
'mandato un messaggio d'errore all'utente.
'In caso contrario verranno chiamate le funzioni "CalcolaCaratteri", "CalcolaEtà" e
'"CalcolaNumero" e la variabile booleana "inserito" verrà posta a "True".
'Da questo momento in poi verrà incrementata la variabile "num" ad ogni click da parte
'dell'utente del Button "Start" a cui seguirà in richiamo della funzione "Controlla" alla
'quale viene passata come parametro la variabile "num".
'Quando la variabile "num" avrà raggiunto il valore 20, verrà reimpostata a -1 e la
'situazione verrà riportata come all'apertura del form.
Private Sub Start_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Start.Click
    If (inserito = False) Then
        If (TextBoxNome.Text <> Nothing) Then
            R.Nome = TextBoxNome.Text
        Else
            MsgBox("Nome non inserito", 1)
            Return
        End If
        If (TextBoxCognome.Text <> Nothing) Then
            R.Cognome = TextBoxCognome.Text
        Else
            MsgBox("Cognome non inserito", 1)
            Return
        End If
        If (TextBoxEtà.Text <> Nothing) Then
            R.Età = TextBoxEtà.Text
        Else
            MsgBox("Età non inserito", 1)
            Return
        End If
        If (TextBoxNumero.Text <> Nothing) Then
            R.Numero = TextBoxNumero.Text
        Else
            MsgBox("Numero non inserito", 1)
            Return
        End If

        TextBoxNome.Enabled = False
        TextBoxCognome.Enabled = False
        TextBoxEtà.Enabled = False
        TextBoxNumero.Enabled = False

        Start.Text = "Continua"
        inserito = True

        CalcolaCaratteri()
        CalcolaEtà()
        CalcolaNumero()

    Else
        num = num + 1
        Controlla(num)

        If (num = 20) Then
            inserito = False
            Start.Text = "Start!"
            num = -1

            TextBoxNome.Enabled = True
            TextBoxCognome.Enabled = True
            TextBoxEtà.Enabled = True
            TextBoxNumero.Enabled = True

            Numera_Label()
        End If
    End If
End Sub

```

La funzione *CalcolaCaratteri* memorizza nell'Array *C* i singoli caratteri inseriti dall'utente nelle **Windows Form TextBox** *TextBoxNome* e *TextBoxCognome*, separati da uno spazio. Se l'intero Array *C* non risulterà pieno, verrà riempito fino alla fine da degli zeri.

Si copia così attraverso un ciclo che va dall'elemento di posizione 0 all'elemento di posizione $n-1$, dove n è la lunghezza della stringa inserita dall'utente nella **Windows Form TextBox** *TextBoxNome*, ogni singolo carattere nell'Array *C*. L'elemento dell'Array *C* di posizione *TextBoxNome.TextLenght*, cioè della lunghezza del valore inserito dall'utente nella **Windows Form TextBox** *TextBoxNome*, sarà uguale al carattere " " (spazio).

Si ripete la stessa operazione per il valore inserito dall'utente nella **Windows Form TextBox** *TextBoxCognome*, tenendo conto questa volta di aggiungere attraverso un ciclo il carattere "0" partendo dall'elemento di posizione *TextBoxNome.TextLenght + 1 + TextBoxCognome.TextLength* fino alla fine dell'Array *C*.

L'Array *C* risulterà così sempre pieno, anche nel peggiore dei casi, e cioè anche quando l'utente decida di inserire una stringa di lunghezza massima sia all'interno della **Windows Form TextBox** *TextBoxNome* che nella **Windows Form TextBox** *TextBoxCognome* in quanto precedentemente è stato imposto, attraverso un controllo, che la stringa inserita dall'utente in entrambe le **Windows Form TextBox** non superi lunghezza 5.

```
'Funzione che memorizza nell'array "C" i caratteri inseriti dall'utente nelle TextBox  
'"TextBoxNome" e "TextBoxCognome" separati da uno spazio. Se l'intero array "C" non  
'risulterà pieno, verrà riempito fino alla fine da degli zeri.
```

```
Private Sub CalcolaCaratteri()
```

```
For i As Integer = 0 To TextBoxNome.TextLength - 1
```

```
    C(i) = TextBoxNome.Text.ElementAt(i)
```

```
Next
```

```
C(TextBoxNome.TextLength) = " "
```

```
For i As Integer = 0 To TextBoxCognome.TextLength - 1
```

```
    C(i + TextBoxNome.TextLength + 1) = TextBoxCognome.Text.ElementAt(i)
```

```
Next
```

```
For i As Integer = (TextBoxNome.TextLength + 1 + TextBoxCognome.TextLength) To 11
```

```
    C(i) = "0"
```

```
Next
```

```
End Sub
```

La funzione *CalcolaEtà* inserisce nell'Array *E* il numero digitato dall'utente nella **Windows Form TextBox** *TextBoxEtà* nel seguente modo: viene trasformato il valore decimale digitato dall'utente in valore binario, tenendo conto che il risultato verrà memorizzato all'interno di un' Array **bidimensionale** *Età* di dimensione 4x8, cioè i primi 8 bit ottenuti come risultato sono registrati nella prima riga dell'Array **bidimensionale**, nella quale l'elemento di posizione (1,0) conterrà il bit di posizione 0 (a partire dal bit meno significativo), l'elemento di posizione (1,1) conterrà il bit di posizione 1, e così via. Nel caso in cui il valore binario ottenuto non occupasse interamente l'Array **bidimensionale** *Età*, allora le celle vuote sono riempite dal valore 0. In questo modo si ottengono 4 byte, che corrispondono alle 4 righe del nostro Array **bidimensionale**. Di ogni singolo byte viene calcolato il suo corrispettivo valore decimale e memorizzato all'interno dell'Array *E*,

seguendo questo procedimento: si calcola il valore decimale della riga x dell'Array **bidimensionale** *Età*, facendo la sommatoria, per y che va da 0 fino a 7, del risultato del prodotto di ogni elemento dell'Array *Età* di posizione y per 2^y . Il risultato della sommatoria viene memorizzato nell'elemento di posizione x dell'Array *E*.

Per la funzione *CalcolaNumero* si esegue lo stesso codice e quindi lo stesso procedimento, tenendo conto però che in questo caso entrano in gioco l'Array **bidimensionale** *Numero* anziché l'Array **bidimensionale** *Età*, l'Array *N* anziché l'Array *E* e la **Windows Form** *TextBox* *TextBoxNumero* anziché la **Windows Form** *TextBox* *TextBoxEtà*.

Funzione *CalcolaEtà*

```
'Funzione che memorizza nell'array "E" il numero inserito dall'utente nella TextBox
'"TextBoxEtà" nel seguente modo: viene trasformato il valore decimale inserito
'dall'utente in valore binario. Tale valore binario viene suddiviso in gruppi da 8 a
'partire dalla cifra meno significativa in modo da ottenere 4 byte. Di ogni singolo
'byte viene calcolato il suo corrispettivo valore decimale e memorizzato all'interno
'dell'array "E". Per questo procedimento viene utilizzato un'array bidimensionale
'"Età" di appoggio.
Private Sub CalcolaEtà()
    'Array bidimensionale di appoggio
    Dim Età(4, 8) As Integer
    'Variabile intera in cui viene memorizzato il valore decimale inserito dall'utente
    'nella TextBox "TextBoxEtà"
    Dim val_età As Integer = TextBoxEtà.Text
    For i As Integer = 0 To 3
        For j As Integer = 0 To 7
            If (val_età <> 0) Then
                Età(i, j) = val_età Mod 2
                val_età = val_età \ 2
            Else
                Età(i, j) = "0"
            End If
        Next
    Next

    For i As Integer = 0 To 3
        val_età = 0
        For j As Integer = 0 To 7
            val_età = val_età + ((2 ^ j) * (Età(i, j)))
        Next
        E(i) = val_età
    Next

End Sub
```


Funzione *CalcolaNumero*

```
'Funzione che memorizza nell'array "N" il numero inserito dall'utente nella TextBox
'"TextBoxNumero" nel seguente modo: viene trasformato il valore decimale inserito
'dall'utente in valore binario. Tale valore binario viene suddiviso in gruppi da 8 a
'partire dalla cifra meno significativa in modo da ottenere 4 byte. Di ogni singolo byte
'viene calcolato il suo corrispettivo valore decimale e memorizzato all'interno
'dell'array "N". Per questo procedimento viene utilizzato un'array bidimensionale "Numero"
Private Sub CalcolaNumero()
    'Array Bidimensionale di appoggio
    Dim Numero(4, 8) As Integer
    'Variabile intera in cui viene memorizzato il valore decimale inserito dall'utente
    'nella TextBox "TextBoxNumero"
    Dim val_num As Integer = TextBoxNumero.Text
    For i As Integer = 0 To 3
        For j As Integer = 0 To 7
            If (val_num <> 0) Then
                Numero(i, j) = val_num Mod 2
                val_num = val_num \ 2
            Else
                Numero(i, j) = "0"
            End If
        Next
    Next

    For i As Integer = 0 To 3
        val_num = 0
        For j As Integer = 0 To 7
            val_num = val_num + ((2 ^ j) * (Numero(i, j)))
        Next
        N(i) = val_num
    Next

End Sub
```

La funziona *Controlla* infine, richiedente un parametro numerico di tipo intero, modifica la proprietà *Text* di due **Windows Form Label**, a secondo del valore del parametro passato. Il valore passato come parametro viene analizzato attraverso uno *switch*: se il valore è compreso tra 0 a 11 allora la proprietà *Text* verrà uguagliata ad un elemento dell'Array *C*, se il valore è compreso tra 12 e 15 allora la proprietà *Text* verrà uguagliata ad un elemento dell'Array *E* mentre se il valore è compreso tra 16 e 19 allora la proprietà *Text* verrà uguagliata ad un elemento dell'Array *N*.

'Funzione che controlla il valore della variabile "num": a seconda del valore di "num",
'verranno modificati solo due Label, ognuno presente in una sola tabella

```
Private Sub Controlla(ByVal num)
    Select Case num
        Case 0
            LabelB0.Text = C(num)
            LabelL0.Text = C(num)
        Case 1
            LabelB1.Text = C(num)
            LabelL1.Text = C(num)
        Case 2
            LabelB2.Text = C(num)
            LabelL2.Text = C(num)
        Case 3
            LabelB3.Text = C(num)
            LabelL3.Text = C(num)
        Case 4
            LabelB4.Text = C(num)
            LabelL4.Text = C(num)
        Case 5
            LabelB5.Text = C(num)
            LabelL5.Text = C(num)
        Case 6
            LabelB6.Text = C(num)
            LabelL6.Text = C(num)
        Case 7
            LabelB7.Text = C(num)
            LabelL7.Text = C(num)
        Case 8
            LabelB8.Text = C(num)
            LabelL8.Text = C(num)
        Case 9
            LabelB9.Text = C(num)
            LabelL9.Text = C(num)
        Case 10
            LabelB10.Text = C(num)
            LabelL10.Text = C(num)
        Case 11
            LabelB11.Text = C(num)
            LabelL11.Text = C(num)
        Case 12
            LabelB12.Text = E(3)
            LabelL12.Text = E(0)
        Case 13
            LabelB13.Text = E(2)
            LabelL13.Text = E(1)
        Case 14
            LabelB14.Text = E(1)
            LabelL14.Text = E(2)
        Case 15
            LabelB15.Text = E(0)
            LabelL15.Text = E(3)
        Case 16
            LabelB16.Text = N(3)
            LabelL16.Text = N(0)
        Case 17
            LabelB17.Text = N(2)
            LabelL17.Text = N(1)
        Case 18
            LabelB18.Text = N(1)
            LabelL18.Text = N(2)
        Case 19
            LabelB19.Text = N(0)
            LabelL19.Text = N(3)
    End Select
End Sub
```

Windows Form Button: *Trasf*

Un'altra funzione che può eseguire il programma è quella di visualizzare il modo in cui un dato, che sia in forma numerica o letterale, purché inserito dall'utente, viene memorizzato all'interno di una macchina *Big-endian* e successivamente trasferito in una macchina *Little-endian*, cioè in particolare la maniera in cui vengono disposti i relativi byte del dato inserito in una macchina *Big-endian* e la maniera in cui vengano disposti i relativi byte del dato inserito se questi venissero trasferiti da una macchina *Big-endian* ad una macchina *Little-endian*.

Facendo un esempio (*Esempio 2.0*):

Considerando l'*Esempio 1.0*, assumiamo che la macchina *Big-endian* spedisca il record a quella *Little-endian*, un byte alla volta, partendo dal byte 0 e terminando con il byte 19 (inoltre assumeremo che non vi siano bit e byte riservati per la trasmissione). Di conseguenza com'è mostrato nella **Figura 2.1**, il byte 0 del calcolatore *Big-endian* va a finire al byte 0 della memoria della macchina *Little-endian*, e così via.

Quando il sistema *Little-endian* prova a stampare il nome e il cognome, tutto funziona correttamente, ma l'età viene trasformata in 21×2^{24} e anche il numero del dipartimento assume un valore diverso. Questo si verifica perché la trasmissione ha invertito, com'è giusto che sia, l'ordine dei caratteri all'interno di una parola, ma allo stesso tempo ha invertito anche i byte all'interno di un intero, cosa che invece non dovrebbe accadere.

Trasferimento da Big-endian a Little-endian

Indirizzo

0

4

8

12

1

Figura 2.1

Questo è il compito del **Windows Form Button** *Trasf*: al suo evento *Click*, viene associato il seguente codice:

```
'Funzione associata all'evento click del button Trasf
Private Sub Trasf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Trasf.Click
    'Viene mostrato il form T
    T.Show()
End Sub
```

Si passa immediatamente alla visualizzazione del Windows Form *T.vb*, attraverso il richiamo della funzione *T.Show()*.

Windows Form : *T.vb*

Il seguente **Windows Form** *T.vb* permette di inserire all'utente dei dati arbitrari e di visualizzare il modo in cui i singoli byte di tali dati vengono disposti all'interno di una macchina *Big-endian* e successivamente visualizzare il modo in cui gli stessi byte vengono disposti all'interno di una macchina *Little-endian*, dopo esser stati trasferiti in quest'ultima a partire da una macchina *Big-endian*.

Nel **Windows Form** *T.vb* vengono impostate nella parte della finestra riguardanti la gestione delle proprietà degli oggetti (ovvero *Proprietà*):

- La dimensione attraverso la proprietà *Size*, con *Width* 784 e *Height* 483, in pixel;
- Lo sfondo attraverso la proprietà *BackgroundImage*, scegliendo un'immagine di sfondo di formato *.jpg* a piacere;
- L'intestazione della form attraverso la proprietà *Text* con “*Trasferimento da Big endian a Little endian*”;
- L'impossibilità di poter ingrandire il **Windows Form** *T.vb* con l'utilizzo del pulsante d'ingrandimento posto nella barra del Menù nell'angolo in alto a destra attraverso la proprietà *MaximizeBox* uguale a *False*;
- L'icona della form attraverso la proprietà *Icon*, scegliendo un'icona di formato *.ico* a piacere;

Dopo aver impostato le caratteristiche del **Windows Form** *T.vb*, vengono introdotti al suo interno alcuni controlli, quali:

- Quattro **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, a cui è stato impostato:
 - Lo sfondo attraverso la proprietà *BackColor* uguale a *Transparent* per rendere lo sfondo dell'oggetto **Windows Form Label** trasparente;
 - L'intestazione attraverso la proprietà *Text* con “*Nome*” per il **Windows Form Label** *LabelNome*, “*Cognome*” per il **Windows Form Label** *LabelCognome*, “*Età*” per il **Windows Form Label** *LabelEtà* e “*Numero reparto*” per il **Windows Form Label** *LabelNumero*;
- Venti **Windows Form Label** (*LabelL0*, *LabelL1*, *LabelL2*, ..., *LabelL19*) e venti **Windows Form Label** (*LabelB0*, *LabelB1*, *LabelB2*, ..., *LabelB19*) all'interno di due **Table Layout Panel**, che visualizzeranno entrambi una numerazione da 0 a 19 inclusi;
- Due **Table Layout Panel**, in cui sarà visualizzata la disposizione dei byte dei dati inseriti dall'utente, ovviamente diversa per ciascun **Table Layout Panel** in quanto uno si riferisce ad una macchina *Big-endian* e l'altro ad una macchina *Little-endian*;
- Quattro **Windows Form TextBox** (*TextBoxNome*, *TextBoxCognome*, *TextBoxEtà* e *TextBoxNumero*) posti uno sotto l'altro accanto, rispettivamente, al **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, per acquisire dall'utente i dati da inserire nella macchina *Big-endian* e *Little-endian*;

- Due **Windows Form Button** *ButtonBigEndian* e *ButtonTrasf*, le cui proprietà *Text* sono rispettivamente “*Big-endian*” e “*Trasferimento*”; in seguito all’evento *Click* del **Windows Form Button** *ButtonBigEndian* verranno disabilitati i quattro **Windows Form TextBox** e il **Windows Form Button** stesso mentre il **Windows Form Button** *ButtonTrasf* verrà abilitato, in quanto risultava disabilitato all’avvio del **Windows Form** *T.vb* e i byte dei dati inseriti dall’utente verranno disposti in maniera diversa all’interno dei due **Table Layout Panel**, per rendere visibile all’utente il meccanismo differente in cui i singoli byte vengono memorizzati all’interno di una macchina *Big-endian* e *Little-endian*, tenendo conto che la disposizione dei byte all’interno della macchina *Little-endian* è dovuta al processo di trasferimento dei byte da una macchina *Big-Endian* ad una macchina *Little-endian*; all’evento *Click* del **Windows Form Button** *ButtonTrasf* verranno abilitati i quattro **Windows Form TextBox** e il **Windows Form Button** *ButtonBigEndian* mentre il **Windows Form Button** stesso verrà disabilitato per permettere all’utente di ricominciare nuovamente il procedimento con dati differenti;
- Un **Windows Form GroupBox**, la cui proprietà *Text* è impostata a *Descrizione* e la proprietà *BackColor* a *Trasparent*, che svolge il ruolo di cornice per contenere al suo interno un **Windows Form Label**, che permetterà all’utente di essere informato attraverso una breve descrizione della specifica sezione del programma che è stata avviata;

In questo modo il **Windows Form** appena creato dell’applicazione presenta il seguente aspetto:

Trasferimento da big-endian a little-endian

Nome:
 Cognome:
 Età:
 Numero reparto:

Big-endian Trasferimento

Big-endian				Indirizzo	Trasferimento da big-endian a little-endian			
0	1	2	3	0	3	2	1	0
4	5	6	7	4	7	6	5	4
8	9	10	11	8	11	10	9	8
12	13	14	15	12	15	14	13	12
16	17	18	19	16	19	18	17	16

Parola a 32 bit Parola a 32 bit

Descrizione

Uno dei problemi più comuni, quando si hanno entrambe le macchine, nasce nel momento in cui una di queste due macchine prova a spedire via rete il record all'altra macchina. In questa sezione del software si assume che la macchina big endian spedisca il record a quella little endian, un byte alla volta, partendo dal byte 0 e terminando con il byte 19 (inoltre, con un certo ottimismo, assumeremo che non vi siano bit e byte riservati per la trasmissione). Di conseguenza, il byte 0 del calcolatore big endian va a finire al byte 0 della memoria della macchina little endian, e così via. Quando il sistema little endian prova a stampare il nome e il cognome tutto funzionerà correttamente, ma l'età e il numero di reparto assumeranno un valore diverso da quello inserito dall'utente. Questa situazione si verifica perché la trasmissione ha invertito, com'è giusto che sia, l'ordine dei caratteri all'interno di una parola, ma ha allo stesso tempo invertito anche i byte all'interno di un intero, cosa che invece non dovrebbe accadere.

Dopo aver creato un nuovo **Windows Form**, è importante soprattutto gestire le risposte dell'applicazione in seguito agli eventi dell'utente o del sistema.

Il primo codice da inserire è quello che riguarda l'evento *Load*, in cui viene impostata la proprietà *Text* dei quattro **Windows Form TextBox** a *Nothing*, viene impostata la proprietà *Enabled* a *False* del **Windows Form Button** *ButtonTrasf* e richiamata infine la funzione *Numera_Label* che inizierà il valore dei 40 **Windows Form Label** per motivi grafici. Quest'ultima funzione è stata già illustrata nella sezione precedente.

```
'Funzione associata all'apertura del form T
Private Sub T_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles MyBase.Load
    TextBoxNome.Text = Nothing
    TextBoxCognome.Text = Nothing
    TextBoxEtà.Text = Nothing
    TextBoxNumero.Text = Nothing

    ButtonTrasf.Enabled = False
    'Richiamo della funzione che inizializza la visualizzazione dei 40 Label all'interno
    'delle due tabelle
    Numera_Label()

End Sub
```

A questo punto bisogna gestire cosa dovrà fare l'applicazione in risposta all'inserimento di un valore all'interno di una **Windows Form TextBox**.

Nel caso di inserimento di un valore all'interno della **Windows Form TextBox** *TextBoxNome* o *TextBoxCognome*, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale e non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

Non tanto diversa è la situazione della **Windows Form TextBox** *TextBoxEtà* o *TextBoxNumero*: nel caso in cui viene inserito un valore, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale. Anche in questo caso il codice non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

La parte più importante di questa sezione che illustra il **Windows Form T.vb** è proprio il codice associato ai due **Windows Form Button**.

Inizialmente l'utente può solo premere il **Windows Form Button** *ButtonBigEndian*, in quanto il **Windows Form Button** *ButtonTrasf* è disabilitato.

Al click del **Windows Form Button** *ButtonBigEndian* si eseguono le seguenti operazioni: per prima cosa bisogna assicurarsi che ogni **Windows Form TextBox** non sia vuota, cioè che la proprietà *Text* sia diversa da "*Nothing*": in questo caso verrà mandato un **MessageBox** d'errore all'utente, altrimenti ogni valore della proprietà *Text* verrà copiato in un corrispondente campo della Struttura Record *R*.

Successivamente le quattro **Windows Form TextBox** vengono disabilitate impostando la loro proprietà *Enabled* a "*False*" come anche viene disabilitato il **Windows Form Button**

ButtonBigEndian impostando la sua proprietà *Enabled* a “False” mentre solo il **Windows Form Button** *ButtonTrasf* viene abilitato impostando la sua proprietà *Enabled* a “True”. La fase successiva è l’esecuzione di tre funzioni : *CalcolaCaratteri*, *CalcolaEtà*, *CalcolaNumero*: queste tre funzioni sono le stesse di quelle utilizzate precedentemente per cui si preferisce saltare il passaggio della spiegazione per non ripeterla.

```
'Funzione associata all'evento click del Button "ButtonBigEndian".
'Se viene cliccato il Button "ButtonBigEndian" ed una delle quattro TextBox è vuota,
'allora viene mandato un messaggio d'errore all'utente.
'In caso contrario verranno chiamate le funzioni "CalcolaCaratteri", "CalcolaEtà" e
'"CalcolaNumero" ed infine la funzione "VisualizzaBigEndian", che servirà per
'visualizzare nella tabella a sinistra l'ordinamento secondo Big-endian
Private Sub ButtonBigEndian_Click(ByVal sender As System.Object, ByVal e As
    System.EventArgs) Handles ButtonBigEndian.Click
    If (TextBoxNome.Text <> Nothing) Then
        R.Nome = TextBoxNome.Text
    Else
        MsgBox("Nome non inserito", 1)
        Return
    End If
    If (TextBoxCognome.Text <> Nothing) Then
        R.Cognome = TextBoxCognome.Text
    Else
        MsgBox("Cognome non inserito", 1)
        Return
    End If
    If (TextBoxEtà.Text <> Nothing) Then
        R.Età = TextBoxEtà.Text
    Else
        MsgBox("Età non inserito", 1)
        Return
    End If
    If (TextBoxNumero.Text <> Nothing) Then
        R.Numero = TextBoxNumero.Text
    Else
        MsgBox("Numero non inserito", 1)
        Return
    End If

    TextBoxNome.Enabled = False
    TextBoxCognome.Enabled = False
    TextBoxEtà.Enabled = False
    TextBoxNumero.Enabled = False

    ButtonBigEndian.Enabled = False
    ButtonTrasf.Enabled = True

    CalcolaCaratteri()
    CalcolaEtà()
    CalcolaNumero()

    'Funzione che serve per visualizzare l'ordinamento secondo Big-endian
    VisualizzaBigEndian()

End Sub
```

Infine per ultima, ma non meno importante, vi è la funzione *VisualizzaBigEndian* che modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel**

a sinistra con uno specifico valore dell'Array *C,E* o *R*. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Big-endian*, guardando la **Tablet Layout Panel** a destra.

'Funzione che serve per visualizzare l'ordinamento secondo Big-endian

Private Sub VisualizzaBigEndian()

```
LabelB0.Text = C(0)
LabelB1.Text = C(1)
LabelB2.Text = C(2)
LabelB3.Text = C(3)
LabelB4.Text = C(4)
LabelB5.Text = C(5)
LabelB6.Text = C(6)
LabelB7.Text = C(7)
LabelB8.Text = C(8)
LabelB9.Text = C(9)
LabelB10.Text = C(10)
LabelB11.Text = C(11)
LabelB12.Text = E(3)
LabelB13.Text = E(2)
LabelB14.Text = E(1)
LabelB15.Text = E(0)
LabelB16.Text = N(3)
LabelB17.Text = N(2)
LabelB18.Text = N(1)
LabelB19.Text = N(0)
```

End Sub

Eseguendo tutte le operazioni fino a qui spiegate, di questa sezione, l'utente si troverà nella condizione di poter solo premere il **Windows Form Button** *ButtonTrasf*.

Con la pressione del **Windows Form Button** *ButtonTrasf* in poche parole l'utente potrà visualizzare in che modo i byte si dispongono in una macchina *Little-endian*, se questi sono stati trasferiti da una macchina *Big-endian*.

All'evento *Click* del **Windows Form Button**, verrà disabilitato lo stesso **Windows Form Button**, impostando la sua proprietà *Enabled* a "*False*", mentre le quattro **Windows Form TextBox** e il **Windows Form Button** *ButtonBigEndian* verranno abilitati, impostando la loro proprietà *Enabled* a "*True*".

Successivamente viene richiamata la funzione *VisualizzaTrasferimento()*, di seguito spiegata.

```

'Funzione associata all'evento click del Button "ButtonTrasf"
'Viene richiamata la funzione "VisualizzaTrasferimento" che servirà per visualizzare
'nella tabella a destra il risultato del trasferimento da una macchina Big-endian ad
'una macchina Little-endian
Private Sub ButtonTrasf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles ButtonTrasf.Click
    ButtonTrasf.Enabled = False
    ButtonBigEndian.Enabled = True

    TextBoxNome.Enabled = True
    TextBoxCognome.Enabled = True
    TextBoxEtà.Enabled = True
    TextBoxNumero.Enabled = True

    'Funzione per serve per visualizzare il trasferimento da una macchina Big-endian
    'ad una macchina Little-endian
    VisualizzaTrasferimento()
End Sub

```

La funzione *VisualizzaTrasferimento* modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel** a destra con uno specifico valore dei Windows Form Label posti nella **Tablet Layout Panel** a sinistra. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Little-endian*, dopo che questi sono stati trasferiti da una macchina *Big-endian*, guardando la **Tablet Layout Panel** a destra.

```

'Funzione per serve per visualizzare il trasferimento da una macchina Big-endian
'ad una macchina Little-endian
Private Sub VisualizzaTrasferimento()
    LabelL0.Text = LabelB0.Text
    LabelL1.Text = LabelB1.Text
    LabelL2.Text = LabelB2.Text
    LabelL3.Text = LabelB3.Text
    LabelL4.Text = LabelB4.Text
    LabelL5.Text = LabelB5.Text
    LabelL6.Text = LabelB6.Text
    LabelL7.Text = LabelB7.Text
    LabelL8.Text = LabelB8.Text
    LabelL9.Text = LabelB9.Text
    LabelL10.Text = LabelB10.Text
    LabelL11.Text = LabelB11.Text
    LabelL12.Text = LabelB12.Text
    LabelL13.Text = LabelB13.Text
    LabelL14.Text = LabelB14.Text
    LabelL15.Text = LabelB15.Text
    LabelL16.Text = LabelB16.Text
    LabelL17.Text = LabelB17.Text
    LabelL18.Text = LabelB18.Text
    LabelL19.Text = LabelB19.Text

End Sub

```

Windows Form Button: *TraInv*

Un'altra funzione che può eseguire il programma è quella di visualizzare il modo in cui un dato, che sia in forma numerica o letterale, purché inserito dall'utente, viene memorizzato all'interno di una macchina *Little-endian*, tenendo conto che tale dato è stato trasferito a partire da una macchina *Big-endian*, e successivamente invertito nella stessa macchina *Little-endian*, tutto questo sotto l'aspetto della disposizione dei byte del dato inserito.

Facendo un esempio (*Esempio 3.0*):

Tenendo conto dell'*Esempio 2.0*, in cui è possibile evincere i problemi nati nel momento in cui una di queste due macchine prova a spedire via rete il record all'altra macchina, si può pensare come prima soluzione quella di invertire i byte all'interno di una parola dopo che la copia è stata effettuata. Facendo in questo modo si otterrebbe quanto mostrato nella *Figura 3.1* dove i due interi sono corretti, mentre la stringa viene trasformata in "MIJTIMS" con la lettera "H" che si perde nel vuoto. L'inversione della stringa si verifica perché il calcolatore, quando la legge, interpreta inizialmente il byte 0 (uno spazio), poi il byte 1(M), e così via.

Trasferimento e inversione

Indirizzo

0

4

8

12

1

Figura 3.1

Questo è il compito del **Windows Form Button** *TraInv*: al suo evento *Click*, viene associato il seguente codice:

```
'Funzione associata all'evento click del Button TraInv
Private Sub TraInv_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles TraInv.Click
    'Viene mostrato il form TI
    TI.Show()
End Sub
```

Si passa immediatamente alla visualizzazione del Windows Form *TI.vb*, attraverso il richiamo della funzione *TI.Show()*.

Windows Form : *TI.vb*

Il seguente **Windows Form** *TI.vb* permette di inserire all'utente dei dati arbitrari e di visualizzare il modo in cui i singoli byte di tali dati vengono disposti all'interno di una macchina *Little-endian*, considerando che sono stati trasferiti da una macchina *Big-endian*, e successivamente visualizzare il modo in cui gli stessi byte vengono disposti all'interno sempre di una macchina *Little-endian*, dopo esser stati invertiti.

Nel **Windows Form** *TI.vb* vengono impostate nella parte della finestra riguardanti la gestione delle proprietà degli oggetti (ovvero *Proprietà*):

- La dimensione attraverso la proprietà *Size*, con *Width* 784 e *Height* 439, in pixel;
- Lo sfondo attraverso la proprietà *BackgroundImage*, scegliendo un'immagine di sfondo di formato *.jpg* a piacere;
- L'intestazione della form attraverso la proprietà *Text* con “*Trasferimento e Inversione*”;
- L'impossibilità di poter ingrandire il **Windows Form** *TI.vb* con l'utilizzo del pulsante d'ingrandimento posto nella barra del Menù nell'angolo in alto a destra attraverso la proprietà *MaximizeBox* uguale a *False*;
- L'icona della form attraverso la proprietà *Icon*, scegliendo un'icona di formato *.ico* a piacere;

Dopo aver impostato le caratteristiche del **Windows Form** *TI.vb*, vengono introdotti al suo interno alcuni controlli, quali:

- Quattro **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, a cui è stato impostato:
 - Lo sfondo attraverso la proprietà *BackColor* uguale a *Transparent* per rendere lo sfondo dell'oggetto **Windows Form Label** trasparente;
 - L'intestazione attraverso la proprietà *Text* con “*Nome*” per il **Windows Form Label** *LabelNome*, “*Cognome*” per il **Windows Form Label** *LabelCognome*, “*Età*” per il **Windows Form Label** *LabelEtà* e “*Numero reparto*” per il **Windows Form Label** *LabelNumero*;
- Venti **Windows Form Label** (*LabelL0*, *LabelL1*, *LabelL2*, ..., *LabelL19*) e venti **Windows Form Label** (*LabelB0*, *LabelB1*, *LabelB2*, ..., *LabelB19*) all'interno di due **Table Layout Panel**, che visualizzeranno entrambi una numerazione da 0 a 19 inclusi;
- Due **Table Layout Panel**, in cui sarà visualizzata la disposizione dei byte dei dati inseriti dall'utente, ovviamente diversa per ciascun **Table Layout Panel** in quanto uno si riferisce ad una macchina *Little-endian*, in cui i byte sono stati solo trasferiti da una macchina *Big-endian*, e l'altro ad una macchina *Little-endian*, in cui i byte sono stati trasferiti e invertiti da una macchina *Big-endian*;
- Quattro **Windows Form TextBox** (*TextBoxNome*, *TextBoxCognome*, *TextBoxEtà* e *TextBoxNumero*) posti uno sotto l'altro accanto, rispettivamente, al **Windows Form**

Label *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, per acquisire dall'utente i dati da inserire nella macchina *Big-endian* e *Little-endian*;

- Due **Windows Form Button** *ButtonTrasf* e *ButtonInv*, le cui proprietà *Text* sono rispettivamente “*Trasferimento*” e “*Inversione*”; in seguito all'evento *Click* del **Windows Form Button** *ButtonTrasf* verranno disabilitati i quattro **Windows Form TextBox** e il **Windows Form Button** stesso mentre il **Windows Form Button** *ButtonInv* verrà abilitato, in quanto risultava disabilitato all'avvio del *Windows Form TI.vb* e i byte dei dati inseriti dall'utente verranno disposti in maniera diversa all'interno dei due **Table Layout Panel**, per rendere visibile all'utente il meccanismo differente in cui i singoli byte vengono memorizzati all'interno di una macchina *Little-endian*, se trasferiti da una macchina *Big-endian*, e sempre all'interno della stessa macchina *Little-endian*, se invertiti; all'evento *Click* del **Windows Form Button** *ButtonInv* verranno abilitati i quattro **Windows Form TextBox** e il **Windows Form Button** *ButtonTrasf* mentre il **Windows Form Button** stesso verrà disabilitato per permettere all'utente di ricominciare nuovamente il procedimento con dati differenti;
- Un **Windows Form GroupBox**, la cui proprietà *Text* è impostata a “*Descrizione*” e la proprietà *BackColor* a “*Trasparent*”, che svolge il ruolo di cornice per contenere al suo interno un **Windows Form Label**, che permetterà all'utente di essere informato attraverso una breve descrizione della specifica sezione del programma che è stata avviata;

In questo modo il **Windows Form** appena creato dell'applicazione presenta il seguente aspetto:

Trasferimento e Inversione

Trasferimento da big-endian a little-endian

Indirizzo

Trasferimento e inversione

Nome

Cognome

Età

Numero reparto

Trasferimento

Inversione

3	2	1	0	0
7	6	5	4	4
11	10	9	8	8
15	14	13	12	12
19	18	17	16	16

Parola a 32 bit

Parola a 32 bit

Descrizione

Una prima soluzione per risolvere il problema del trasferimento di un record da una macchina big endian ad una macchina little endian, potrebbe essere quella di usare un programma che inverte i byte all'interno di una parola dopo che la copia è stata effettuata.

Facendo in questo modo si otterrebbe che i due interi sono corretti, mentre la stringa, composta dal nome e cognome, si trasforma perdendo il suo valore iniziale inserito dall'utente. L'inversione della stringa si verifica perchè il calcolatore, quando la legge, interpreta inizialmente il byte 0, poi il byte 1, e così via.

Dopo aver creato un nuovo **Windows Form**, è importante soprattutto gestire le risposte dell'applicazione in seguito agli eventi dell'utente o del sistema.

Il primo codice da inserire è quello che riguarda l'evento *Load*, in cui viene impostata la proprietà *Text* dei quattro **Windows Form TextBox** a “*Nothing*”, viene impostata la proprietà *Enabled* a *False* del **Windows Form Button** *ButtonInv* e richiamata infine la funzione *Numera_Label* che inizierà il valore dei 40 **Windows Form Label** per motivi grafici. Quest'ultima funzione è stata già illustrata nella sezione precedente.

```
'Funzione associata all'apertura del form TI
Private Sub TI_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles MyBase.Load
    TextBoxNome.Text = Nothing
    TextBoxCognome.Text = Nothing
    TextBoxEtà.Text = Nothing
    TextBoxNumero.Text = Nothing

    ButtonInv.Enabled = False
    'Richiamo della funzione che inizializza la visualizzazione dei 40 Label all'interno
    'delle due tabelle
    Numera_Label()

End Sub
```

A questo punto bisogna gestire cosa dovrà fare l'applicazione in risposta all'inserimento di un valore all'interno di una **Windows Form TextBox**.

Nel caso di inserimento di un valore all'interno della **Windows Form TextBox** *TextBoxNome* o *TextBoxCognome*, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale e non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

Non tanto diversa è la situazione della **Windows Form TextBox** *TextBoxEtà* o *TextBoxNumero*: nel caso in cui viene inserito un valore, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale. Anche in questo caso il codice non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

La parte più importante di questa sezione che illustra il **Windows Form TI.vb** è proprio il codice associato ai due **Windows Form Button**.

Inizialmente l'utente può solo premere il **Windows Form Button** *ButtonTrasf*, in quanto il **Windows Form Button** *ButtonInv* è disabilitato.

Al click del **Windows Form Button** *ButtonTrasf* si eseguono le seguenti operazioni: per prima cosa bisogna assicurarsi che ogni **Windows Form TextBox** non sia vuota, cioè che la proprietà *Text* sia diversa da “*Nothing*”: in questo caso verrà mandato un **MessageBox** d'errore all'utente, altrimenti ogni valore della proprietà *Text* verrà copiato in un corrispondente campo della Struttura Record *R*.

Successivamente le quattro **Windows Form TextBox** vengono disabilite impostando la loro proprietà *Enabled* a “*False*” come anche viene disabilitato il **Windows Form Button** *ButtonTrasf* impostando la sua proprietà *Enabled* a “*False*” mentre solo il **Windows Form Button** *ButtonInv* viene abilitato impostando la sua proprietà *Enabled* a “*True*”.

La fase successiva è l'esecuzione di tre funzioni : *CalcolaCaratteri*, *CalcolaEtà*, *CalcolaNumero*: queste tre funzioni sono le stesse di quelle utilizzate precedentemente per cui si preferisce saltare il passaggio della spiegazione per non ripeterla.

```
'Funzione associata all'evento click del Button "ButtonTrasf".
'Se viene cliccato il Button "ButtonTrasf" ed una delle quattro TextBox è vuota, allora
'viene mandato un messaggio d'errore all'utente.
'In caso contrario verranno chiamate le funzioni "CalcolaCaratteri", "CalcolaEtà" e
'"CalcolaNumero" ed infine la funzione "VisualizzaTrasferimento", che servirà per
'visualizzare nella tabella a sinistra il risultato del trasferimento da una macchina
'Big-endian ad una macchina Little-endian.
Private Sub ButtonTrasf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles ButtonTrasf.Click
    If (TextBoxNome.Text <> Nothing) Then
        R.Nome = TextBoxNome.Text
    Else
        MsgBox("Nome non inserito", 1)
        Return
    End If
    If (TextBoxCognome.Text <> Nothing) Then
        R.Cognome = TextBoxCognome.Text
    Else
        MsgBox("Cognome non inserito", 1)
        Return
    End If
    If (TextBoxEtà.Text <> Nothing) Then
        R.Età = TextBoxEtà.Text
    Else
        MsgBox("Età non inserito", 1)
        Return
    End If
    If (TextBoxNumero.Text <> Nothing) Then
        R.Numero = TextBoxNumero.Text
    Else
        MsgBox("Numero non inserito", 1)
        Return
    End If

    TextBoxNome.Enabled = False
    TextBoxCognome.Enabled = False
    TextBoxEtà.Enabled = False
    TextBoxNumero.Enabled = False

    ButtonInv.Enabled = True
    ButtonTrasf.Enabled = False

    CalcolaCaratteri()
    CalcolaEtà()
    CalcolaNumero()

    'Funzione per serve per visualizzare il trasferimento da una macchina Big-endian
    'ad una macchina Little-endian
    VisualizzaTrasferimento()

End Sub
```

Infine per ultima, ma non meno importante, vi è la funzione *VisualizzaTrasferimento* che modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel** a sinistra con uno specifico valore dell'Array *C,E* o *R*. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Little-endian*, tenendo conto che hanno subito il processo di trasferimento da una macchina *Big-endian*.

```

'Funzione per serve per visualizzare il trasferimento da una macchina Big-endian
'ad una macchina Little-endian
Private Sub VisualizzaTrasferimento()
    LabelB0.Text = C(3)
    LabelB1.Text = C(2)
    LabelB2.Text = C(1)
    LabelB3.Text = C(0)
    LabelB4.Text = C(7)
    LabelB5.Text = C(6)
    LabelB6.Text = C(5)
    LabelB7.Text = C(4)
    LabelB8.Text = C(11)
    LabelB9.Text = C(10)
    LabelB10.Text = C(9)
    LabelB11.Text = C(8)
    LabelB12.Text = E(0)
    LabelB13.Text = E(1)
    LabelB14.Text = E(2)
    LabelB15.Text = E(3)
    LabelB16.Text = N(0)
    LabelB17.Text = N(1)
    LabelB18.Text = N(2)
    LabelB19.Text = N(3)
End Sub

```

Eseguendo anche questa volta tutte le operazioni fino a qui spiegate, di questa sezione, l'utente si troverà nella condizione di poter solo premere il **Windows Form Button ButtonInv**.

Con la pressione del **Windows Form Button ButtonInv** in poche parole l'utente potrà visualizzare la disposizione invertita dei byte, ottenuti dall'operazione precedente, sempre in una macchina *Little-endian*.

All'evento *Click* del **Windows Form Button**, verrà disabilitato lo stesso **Windows Form Button**, impostando la sua proprietà *Enabled* a "*False*", mentre le quattro **Windows Form TextBox** e il **Windows Form Button ButtonTrasf** verranno abilitati, impostando la loro proprietà *Enabled* a "*True*".

Successivamente viene richiamata la funzione *VisualizzaInversione()*, di seguito spiegata.

```

'Funzione associata all'evento click del Button "ButtonInversione".
'Viene richiamata la funzione "VisualizzaInversione", che servirà per visualizzare
'nella tabella a destra il risultato dell'inversione dei byte nella tabella a destra.
Private Sub ButtonInversione_Click(ByVal sender As System.Object, ByVal e _
    As System.EventArgs) Handles ButtonInv.Click
    ButtonTrasf.Enabled = True
    ButtonInv.Enabled = False

    TextBoxNome.Enabled = True
    TextBoxCognome.Enabled = True
    TextBoxEtà.Enabled = True
    TextBoxNumero.Enabled = True

    'Funzione che visualizza il risultato dell'inversione dei byte nella tabella a destra.
    VisualizzaInversione()
End Sub

```


La funzione *VisualizzaInversione* modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel** a destra con uno specifico valore dei Windows Form Label posti nella Tablet Layout Panel a sinistra. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Little-endian*, dopo che questi sono stati trasferiti da una macchina *Big-endian* e invertiti.

```
'Funzione che visualizza il risultato dell'inversione dei byte nella tabella a destra.
```

```
Private Sub VisualizzaInversione()  
    LabelL0.Text = LabelB0.Text  
    LabelL1.Text = LabelB1.Text  
    LabelL2.Text = LabelB2.Text  
    LabelL3.Text = LabelB3.Text  
    LabelL4.Text = LabelB4.Text  
    LabelL5.Text = LabelB5.Text  
    LabelL6.Text = LabelB6.Text  
    LabelL7.Text = LabelB7.Text  
    LabelL8.Text = LabelB8.Text  
    LabelL9.Text = LabelB9.Text  
    LabelL10.Text = LabelB10.Text  
    LabelL11.Text = LabelB11.Text  
    LabelL12.Text = LabelB12.Text  
    LabelL13.Text = LabelB13.Text  
    LabelL14.Text = LabelB14.Text  
    LabelL15.Text = LabelB15.Text  
    LabelL16.Text = LabelB16.Text  
    LabelL17.Text = LabelB17.Text  
    LabelL18.Text = LabelB18.Text  
    LabelL19.Text = LabelB19.Text  
End Sub
```

Windows Form Button: Sol

Ultima ma non meno importante è la parte riguardante una soluzione personale ai problemi nati nel trasferimento di dati da una macchina all'altra per via delle differenti convenzioni sull'ordinamento dei byte. La funzione quindi che può eseguire il programma è quella di visualizzare il modo in cui un dato, che sia in forma numerica o letterale, purché inserito dall'utente, viene memorizzato all'interno di una macchina *Big-endian* e successivamente trasferito in una macchina *Little-endian*, eliminando i problemi nati nel trasferimento.

Facendo un esempio (*Esempio 4.0*):

Facendo riferimento all'*Esempio 2.0*, in cui si evincono chiaramente i problemi nati nel momento in cui una di queste due macchine prova a spedire via rete il record all'altra macchina, un buon metodo, ben poco efficiente, è quello di includere un'intestazione prima di ogni dato per specificarne il tipo (stringa, intero o altro) in modo che il ricevente possa effettuare solo le conversioni realmente necessarie, come mostrato il *Figura 4.1* :

Big-endian**Indirizzo****Soluzione personale**

0

4

8

12

16

	M	I	J
T	I	M	S
0	0	0	H
0	0	0	21
0	0	1	4

Figura 4.1

Nel nostro caso non è stato sufficiente far specificare all'utente il tipo di dato inserito in quanto è stato possibile conoscerlo a priori.

Questo è il compito del **Windows Form Button Sol**: al suo evento *Click*, viene associato il seguente codice:

```
'Funzione associata all'evento click del Button Sol
Private Sub Sol_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles Sol.Click
    'Viene mostrato il form S
    S.Show()
End Sub
```

Si passa immediatamente alla visualizzazione del Windows Form *S.vb*, attraverso il richiamo della funzione *S.Show()*.

Windows Form : S.vb

Il seguente **Windows Form S.vb** permette di inserire all'utente dei dati arbitrari e di visualizzare il modo in cui i singoli byte di tali dati vengono disposti all'interno di una macchina *Big-endian* e successivamente visualizzare il modo in cui gli stessi byte vengono disposti all'interno di una macchina *Little-endian*, dopo esser stati trasferiti e risolvendo i problemi che nascono proprio nel processo di trasferimento.

Nel **Windows Form S.vb** vengono impostate nella parte della finestra riguardanti la gestione delle proprietà degli oggetti (ovvero *Proprietà*):

- La dimensione attraverso la proprietà *Size*, con *Width* 784 e *Height* 418, in pixel;
- Lo sfondo attraverso la proprietà *BackgroundImage*, scegliendo un'immagine di sfondo di formato *.jpg* a piacere;
- L'intestazione della form attraverso la proprietà *Text* con "Soluzione";
- L'impossibilità di poter ingrandire il **Windows Form S.vb** con l'utilizzo del pulsante d'ingrandimento posto nella barra del Menù nell'angolo in alto a destra attraverso la proprietà *MaximizeBox* uguale a *False*;
- L'icona della form attraverso la proprietà *Icon*, scegliendo un'icona di formato *.ico* a piacere;

Dopo aver impostato le caratteristiche del Windows Form *S.vb*, vengono introdotti al suo interno alcuni controlli, quali:

- Quattro **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, a cui è stato impostato:
 - Lo sfondo attraverso la proprietà *BackColor* uguale a *Transparent* per rendere lo sfondo dell'oggetto **Windows Form Label** trasparente;
 - L'intestazione attraverso la proprietà *Text* con “*Nome*” per il **Windows Form Label** *LabelNome*, “*Cognome*” per il **Windows Form Label** *LabelCognome*, “*Età*” per il **Windows Form Label** *LabelEtà* e “*Numero reparto*” per il **Windows Form Label** *LabelNumero*;
- Venti **Windows Form Label** (*LabelL0*, *LabelL1*, *LabelL2*, ..., *LabelL19*) e venti **Windows Form Label** (*LabelB0*, *LabelB1*, *LabelB2*, ..., *LabelB19*) all'interno di due **Table Layout Panel**, che visualizzeranno entrambi una numerazione da 0 a 19 inclusi;
- Due **Table Layout Panel**, in cui sarà visualizzata la disposizione dei byte dei dati inseriti dall'utente, ovviamente diversa per ciascun **Table Layout Panel** in quanto uno si riferisce ad una macchina *Big-endian* e l'altro ad una macchina *Little-endian*;
- Quattro **Windows Form TextBox** (*TextBoxNome*, *TextBoxCognome*, *TextBoxEtà* e *TextBoxNumero*) posti uno sotto l'altro accanto, rispettivamente, al **Windows Form Label** *LabelNome*, *LabelCognome*, *LabelEtà* e *LabelNumero*, per acquisire dall'utente i dati da inserire nella macchina *Big-endian* e *Little-endian*;
- Due **Windows Form Button** *ButtonBigEndian* e *ButtonTrasf*, le cui proprietà *Text* sono rispettivamente “*Big-endian*” e “*Trasferimento*”; in seguito all'evento *Click* del **Windows Form Button** *ButtonBigEndian* verranno disabilitati i quattro **Windows Form TextBox** e il **Windows Form Button** stesso mentre il **Windows Form Button** *ButtonTrasf* verrà abilitato, in quanto risultava disabilitato all'avvio del **Windows Form S.vb** e i byte dei dati inseriti dall'utente verranno disposti in maniera diversa all'interno dei due **Table Layout Panel**, per rendere visibile all'utente il meccanismo differente in cui i singoli byte vengono memorizzati all'interno di una macchina *Big-endian* e *Little-endian*, tenendo conto che la disposizione dei byte all'interno della macchina *Little-endian* è dovuta al processo di “trasferimento” dei byte da una macchina *Big-Endian* ad una macchina *Little-endian*; all'evento *Click* del **Windows Form Button** *ButtonTrasf* verranno abilitati i quattro **Windows Form TextBox** e il **Windows Form Button** *ButtonBigEndian* mentre il **Windows Form Button** stesso verrà disabilitato per permettere all'utente di ricominciare nuovamente il procedimento con dati differenti;
- Un **Windows Form GroupBox**, la cui proprietà *Text* è impostata a “*Descrizione*” e la proprietà *BackColor* a “*Transparent*”, che svolge il ruolo di cornice per contenere al suo interno un **Windows Form Label**, che permetterà all'utente di essere informato attraverso una breve descrizione della specifica sezione del programma che è stata avviata;

In questo modo il **Windows Form** appena creato dell'applicazione presenta il seguente aspetto:

Soluzione

Nome

Cognome

Età

Numero reparto

Big-endian				Indirizzo	Trasferimento da big-endian a little-endian			
0	1	2	3	0	3	2	1	0
4	5	6	7	4	7	6	5	4
8	9	10	11	8	11	10	9	8
12	13	14	15	12	15	14	13	12
16	17	18	19	16	19	18	17	16

Parola a 32 bit

Descrizione

Non esiste una soluzione semplice al problema del trasferimento di un record da una macchina big endian ad una macchina little endian. Un metodo, ben poco efficiente, è quello di includere un'intestazione prima di ogni dato per specificarne il tipo (stringa, intero o altro) e la lunghezza, in modo che il ricevente possa effettuare solo le conversioni realmente necessarie. In ogni caso dovrebbe essere chiaro che la mancanza di uno standard per l'ordinamento dei byte costituisce una grande scomodità nello scambio di dati fra macchine di tipo diverso.

Dopo aver creato un nuovo **Windows Form**, è importante soprattutto gestire le risposte dell'applicazione in seguito agli eventi dell'utente o del sistema.

Il primo codice da inserire è quello che riguarda l'evento *Load*, in cui viene impostata la proprietà *Text* dei quattro **Windows Form TextBox** a "*Nothing*", viene impostata la proprietà *Enabled* a "*False*" del **Windows Form Button** *ButtonTrasf* e richiamata infine la funzione *Numera_Label* che inizierà il valore dei 40 **Windows Form Label** per motivi grafici. Quest'ultima funzione è stata già illustrata nella sezione precedente.

```
'Funzione associata all'apertura del form S
Private Sub S_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) _
    Handles MyBase.Load
    TextBoxNome.Text = Nothing
    TextBoxCognome.Text = Nothing
    TextBoxEtà.Text = Nothing
    TextBoxNumero.Text = Nothing

    ButtonTrasf.Enabled = False
    'Richiamo della funzione che inizializza la visualizzazione dei 40 Label all'interno
    'delle due tabelle
    Numera_Label()
End Sub
```

A questo punto bisogna gestire cosa dovrà fare l'applicazione in risposta all'inserimento di un valore all'interno di una **Windows Form TextBox**.

Nel caso di inserimento di un valore all'interno della **Windows Form TextBox** *TextBoxNome* o *TextBoxCognome*, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale e non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

Non tanto diversa è la situazione della **Windows Form TextBox** *TextBoxEtà* o *TextBoxNumero*: nel caso in cui viene inserito un valore, l'applicazione risponde allo stesso modo, cioè il codice associato all'evento *KeyPress* di entrambe le **Windows Form TextBox** è uguale. Anche in questo caso il codice non verrà trattato in quanto è lo stesso di quello spiegato precedentemente.

La parte più importante di questa sezione che illustra il **Windows Form T.vb** è proprio il codice associato ai due **Windows Form Button**.

Inizialmente l'utente può solo premere il **Windows Form Button** *ButtonBigEndian*, in quanto il **Windows Form Button** *ButtonTrasf* è disabilitato.

Al click del **Windows Form Button** *ButtonBigEndian* si eseguono le seguenti operazioni: per prima cosa bisogna assicurarsi che ogni **Windows Form TextBox** non sia vuota, cioè che la proprietà *Text* sia diversa da “*Nothing*”: in questo caso verrà mandato un **MessageBox** d'errore all'utente, altrimenti ogni valore della proprietà *Text* verrà copiato in un corrispondente campo della Struttura Record *R*.

Successivamente le quattro **Windows Form TextBox** vengono disabilite impostando la loro proprietà *Enabled* a “*False*” come anche viene disabilitato il **Windows Form Button** *ButtonBigEndian* impostando la sua proprietà *Enabled* a “*False*” mentre solo il **Windows Form Button** *ButtonTrasf* viene abilitato impostando la sua proprietà *Enabled* a “*True*”.

La fase successiva è l'esecuzione di tre funzioni : *CalcolaCaratteri*, *CalcolaEtà*, *CalcolaNumero*: queste tre funzioni sono le stesse di quelle utilizzate precedentemente per cui si preferisce saltare il passaggio della spiegazione per non ripeterla.

```

'Funzione associata all'evento click del Button "ButtonBigEndian".
'Se viene cliccato il Button "ButtonBigEndian" ed una delle quattro TextBox è vuota,
' allora viene mandato un messaggio d'errore all'utente.
'In caso contrario verranno chiamate le funzioni "CalcolaCaratteri", "CalcolaEtà" e
'"CalcolaNumero" ed infine la funzione "VisualizzaBigEndian", che servirà per
'visualizzare nella tabella a sinistra l'ordinamento secondo Big-endian
Private Sub ButtonBigEndian_Click(ByVal sender As System.Object, ByVal e _
    As System.EventArgs) Handles ButtonBigEndian.Click
    If (TextBoxNome.Text <> Nothing) Then
        R.Nome = TextBoxNome.Text
    Else
        MsgBox("Nome non inserito", 1)
        Return
    End If
    If (TextBoxCognome.Text <> Nothing) Then
        R.Cognome = TextBoxCognome.Text
    Else
        MsgBox("Cognome non inserito", 1)
        Return
    End If
    If (TextBoxEtà.Text <> Nothing) Then
        R.Età = TextBoxEtà.Text
    Else
        MsgBox("Età non inserito", 1)
        Return
    End If
    If (TextBoxNumero.Text <> Nothing) Then
        R.Numero = TextBoxNumero.Text
    Else
        MsgBox("Numero non inserito", 1)
        Return
    End If

    TextBoxNome.Enabled = False
    TextBoxCognome.Enabled = False
    TextBoxEtà.Enabled = False
    TextBoxNumero.Enabled = False

    ButtonTrasf.Enabled = True
    ButtonBigEndian.Enabled = False

    CalcolaCaratteri()
    CalcolaEtà()
    CalcolaNumero()
    'Funzione che serve per visualizzare l'ordinamento secondo Big-endian
    VisualizzaBigEndian()

End Sub

```

Infine per ultima, ma non meno importante, vi è la funzione *VisualizzaBigEndian* che modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel** a sinistra con uno specifico valore dell'Array *C,E* o *R*. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Big-endian*, guardando la **Tablet Layout Panel** a destra.

```

'Funzione che serve per visualizzare l'ordinamento secondo Big-endian
Private Sub VisualizzaBigEndian()
    LabelB0.Text = C(0)
    LabelB1.Text = C(1)
    LabelB2.Text = C(2)
    LabelB3.Text = C(3)
    LabelB4.Text = C(4)
    LabelB5.Text = C(5)
    LabelB6.Text = C(6)
    LabelB7.Text = C(7)
    LabelB8.Text = C(8)
    LabelB9.Text = C(9)
    LabelB10.Text = C(10)
    LabelB11.Text = C(11)
    LabelB12.Text = E(3)
    LabelB13.Text = E(2)
    LabelB14.Text = E(1)
    LabelB15.Text = E(0)
    LabelB16.Text = N(3)
    LabelB17.Text = N(2)
    LabelB18.Text = N(1)
    LabelB19.Text = N(0)

End Sub

```

Eseguendo tutte le operazioni fino a qui spiegate, di questa sezione, l'utente si troverà nella condizione di poter solo premere il **Windows Form Button** *ButtonTrasf*.

Con la pressione del **Windows Form Button** *ButtonTrasf* in poche parole l'utente potrà visualizzare in che modo i byte si dispongono in una macchina *Little-endian*, se questi sono stati trasferiti da una macchina *Big-endian*, tenendo conto anche di risolvere i problemi dovuti al trasferimento.

All'evento *Click* del **Windows Form Button**, verrà disabilitato lo stesso **Windows Form Button**, impostando la sua proprietà *Enabled* a "*False*", mentre le quattro **Windows Form TextBox** e il **Windows Form Button** *ButtonBigEndian* verranno abilitati, impostando la loro proprietà *Enabled* a "*True*".

Successivamente viene richiamata la funzione *VisualizzaSoluzione()*, di seguito spiegata.

```

'Funzione associata all'evento click del Button "ButtonTrasferimento".
Private Sub ButtonTrasferimento_Click(ByVal sender As System.Object, ByVal e _
    As System.EventArgs) Handles ButtonTrasf.Click
    ButtonBigEndian.Enabled = True
    ButtonTrasf.Enabled = False

    TextBoxNome.Enabled = True
    TextBoxCognome.Enabled = True
    TextBoxEtà.Enabled = True
    TextBoxNumero.Enabled = True

    'Funzione che serve per visualizzare la soluzione personale sul trasferimento
    'dei byte da una macchina Big-endian ad una macchina Little-endian
    VisualizzaSoluzione()

End Sub

```

La funzione *VisualizzaSoluzione* modifica la proprietà *Text* di tutti i **Windows Form Label** posti nella **Tablet Layout Panel** a destra con uno specifico valore dei Windows Form Label posti nella **Tablet Layout Panel** a sinistra. In questo modo l'utente potrà visualizzare in che maniera i byte dei dati inseriti vengono disposti in una macchina *Little-endian*, dopo che questi sono stati trasferiti da una macchina *Big-endian*, guardando la **Tablet Layout Panel** a destra. Potrà inoltre notare che non saranno presenti i problemi dovuti al trasferimento.

```
'Funzione che serve per visualizzare la soluzione personale sul trasferimento  
'dei byte da una macchina Big-endian ad una macchina Little-endian  
Private Sub VisualizzaSoluzione()
```

```
LabelL0.Text = LabelB0.Text  
LabelL1.Text = LabelB1.Text  
LabelL2.Text = LabelB2.Text  
LabelL3.Text = LabelB3.Text  
LabelL4.Text = LabelB4.Text  
LabelL5.Text = LabelB5.Text  
LabelL6.Text = LabelB6.Text  
LabelL7.Text = LabelB7.Text  
LabelL8.Text = LabelB8.Text  
LabelL9.Text = LabelB9.Text  
LabelL10.Text = LabelB10.Text  
LabelL11.Text = LabelB11.Text  
LabelL12.Text = LabelB15.Text  
LabelL13.Text = LabelB14.Text  
LabelL14.Text = LabelB13.Text  
LabelL15.Text = LabelB12.Text  
LabelL16.Text = LabelB19.Text  
LabelL17.Text = LabelB18.Text  
LabelL18.Text = LabelB17.Text  
LabelL19.Text = LabelB16.Text
```

```
End Sub
```

Idee

E' possibile attuare delle possibile modifiche per migliorare le prestazioni e le funzionalità del programma, quali ad esempio quelle elencate nei 3 punti seguenti:

- Possibilità di inserire una stringa di lunghezza superiore a 5 all'interno delle **Windows Form TextBox** *TextBoxNome* e *TextBoxCognome*; per motivi puramente grafici, è stato imposto all'utente di poter inserire soltanto massimo 5 caratteri sia per il nome che per il cognome, in quanto la parte letterale del risultato delle operazioni, che verrà visualizzata nella tabella, può occupare massimo 3 righe, cioè 12 byte (4 byte per riga). In questo modo nel peggiore dei casi non ci saranno problemi di visualizzazione, tenendo conto anche del byte per lo spazio e del byte a fine stringa. Un miglioramento potrebbe essere appunto quello di eliminare questo limite per l'utente, considerando che il numero di righe della tabella, in cui verrà visualizzato il risultato, cambierà di volta in volta in maniera dinamica.

- Possibilità di controllare il valore numerico inserito dall'utente all'interno delle **Windows Form TextBox** *TextBoxEtà* e *TextBoxNumero*; non è stato gestito il massimo valore numerico consentito all'utente di poter inserire, sia nel caso dell'età che nel caso del numero (il discorso è identico). Tenendo conto che soltanto una riga della tabella è adibita per la parte numerica del risultato, ciò significa che l'utente può inserire al massimo un valore numerico pari al corrispettivo valore decimale del numero in binario composto da tanti 1 quanti sono i bit a disposizione, ovvero 32. Un miglioramento potrebbe appunto essere quello di imporre questo limite per l'utente;
- Possibilità di ottenere il risultato di un'eventuale stampa dei sistemi *Big-endian* e *Little-endian* che vengono utilizzati per ogni funzionalità del programma; quando viene eseguita una funzionalità del programma, si ottiene in maniera visiva attraverso una tabella la disposizione dei byte del risultato di tale funzionalità, ma non si ottiene una visualizzazione del risultato sotto forma di sequenza di byte. Anche questo potrebbe essere un miglioramento da tenere in considerazione per aumentare l'efficienza del programma: per capire meglio, si potrebbe ottenere, ad esempio, sotto la tabella referente il risultato del trasferimento dei byte da una macchina *Big-endian* ad una macchina *Little-endian*, la sequenza ordinata dei byte ottenuti nella macchina *Little-endian*, secondo l'ordinamento appunto *Little-endian*;
- Possibilità di trattare anche trasferimenti e conversioni nell'altra direzione, ovvero da *Little-Endian* a *Big-Endian* dando per esempio la possibilità all'utente, all'inizio dell'avvio dell'eseguibile, di poter scegliere quale direzione di trasferimento e conversione far fare al programma.