

Introduzione a Visual Basic

Lezione 2

Cicli e anomalie

Mario Alviano

Introduzione all'informatica
Università della Calabria

<http://alviano.net/introinfo>

A.A. 2008/09

Introduzione

- **Finora abbiamo visto programmi molto semplici**
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i cicli

Introduzione

- Finora abbiamo visto programmi molto semplici
- Sequenze di istruzioni lineari
 - Al più qualche istruzione condizionale
- La dimensione dell'input era nota a priori
 - Ad esempio, nel programma per il calcolo della media abbiamo fissato 5 valori
- Nel mondo reale, questo tipo di programmi serve a poco
- Possiamo generalizzare programmi di questo tipo a un numero arbitrario di valori?
 - Certamente! Usando i **cicli**

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (1)

- Come fareste, a mente, la somma dei seguenti numeri?
 - 13, 16, 4, ...
- Probabilmente iniziereste pensando: “La somma di ZERO numeri è 0”
- Vedendo il 13 penserete: “Alla somma precedente (0) devo aggiungere 13. Quindi, ottengo 13”
- Ora trovate 16: “Alla somma precedente (13) devo aggiungere 16. Quindi, ottengo 29”
- Trovando il 4: “Alla somma precedente (29) devo aggiungere 4. Quindi, ottengo 33”
- ...
- **Nota:** viene ripetuta sempre la stessa operazione
 - Sommare il nuovo numero ai precedenti
- Un ciclo è proprio questo!

Un esempio concreto: sommare una sequenza (2)

- Un ciclo è un blocco di istruzioni che viene ripetuto finché una certa condizione è vera
- Ad esempio, una struttura del tipo

```
Mentre CONDIZIONE
  BLOCCO ISTRUZIONI
Fine Mentre
```
- BLOCCO ISTRUZIONI viene eseguito se CONDIZIONE è vera
- A differenza dell'istruzione condizionale (Se), però, CONDIZIONE viene ricontrollata al termine del BLOCCO ISTRUZIONI
- Se è ancora vera, BLOCCO ISTRUZIONI viene ripetuto
- ...

Un esempio concreto: sommare una sequenza (2)

- Un ciclo è un blocco di istruzioni che viene ripetuto finché una certa condizione è vera
- Ad esempio, una struttura del tipo
 - Mentre** CONDIZIONE
 - BLOCCO ISTRUZIONI
 - Fine Mentre**
- BLOCCO ISTRUZIONI viene eseguito se CONDIZIONE è vera
- A differenza dell'istruzione condizionale (Se), però, CONDIZIONE viene ricontrollata al termine del BLOCCO ISTRUZIONI
- Se è ancora vera, BLOCCO ISTRUZIONI viene ripetuto
- ...

Un esempio concreto: sommare una sequenza (2)

- Un ciclo è un blocco di istruzioni che viene ripetuto finché una certa condizione è vera
- Ad esempio, una struttura del tipo
 - Mentre CONDIZIONE
 - BLOCCO ISTRUZIONI
 - Fine Mentre
- BLOCCO ISTRUZIONI viene eseguito se CONDIZIONE è vera
- A differenza dell'istruzione condizionale (Se), però, CONDIZIONE viene ricontrollata al termine del BLOCCO ISTRUZIONI
- Se è ancora vera, BLOCCO ISTRUZIONI viene ripetuto
- ...

Un esempio concreto: sommare una sequenza (2)

- Un ciclo è un blocco di istruzioni che viene ripetuto finché una certa condizione è vera
- Ad esempio, una struttura del tipo
 - Mentre CONDIZIONE
 - BLOCCO ISTRUZIONI
 - Fine Mentre
- BLOCCO ISTRUZIONI viene eseguito se CONDIZIONE è vera
- A differenza dell'istruzione condizionale (Se), però, CONDIZIONE viene ricontrollata al termine del BLOCCO ISTRUZIONI
- Se è ancora vera, BLOCCO ISTRUZIONI viene ripetuto
- ...

Un esempio concreto: sommare una sequenza (2)

- Un ciclo è un blocco di istruzioni che viene ripetuto finché una certa condizione è vera
- Ad esempio, una struttura del tipo
 - Mentre CONDIZIONE
 - BLOCCO ISTRUZIONI
 - Fine Mentre
- BLOCCO ISTRUZIONI viene eseguito se CONDIZIONE è vera
- A differenza dell'istruzione condizionale (Se), però, CONDIZIONE viene ricontrollata al termine del BLOCCO ISTRUZIONI
- Se è ancora vera, BLOCCO ISTRUZIONI viene ripetuto
- ...

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1?

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1? OK!

Un esempio concreto: sommare una sequenza (3)

- Cosa ci serve per sommare una sequenza di numeri?
- Sicuramente UNA variabile per mantenere la somma (parziale)
- Ci serve una variabile per ogni numero nella sequenza?
 - No... ci basta leggere un numero per volta e sommarlo alla somma precedente
- Ma quando termina la sequenza?
 - Possiamo usare un numero specifico per indicare che la sequenza è finita
 - Scegliamo -1? OK! Ma potevamo scegliere un qualsiasi altro numero

Un esempio concreto: sommare una sequenza (4)

Esempio (Calcolo della somma: meta-programma)

```
Programma principale()
  Variabili: x, somma

  somma = 0

  x = Leggi("Inserisci un numero, -1 per terminare")
  Mentre x è diverso da -1
    somma = somma + x
    x = Leggi("Inserisci un numero, -1 per terminare")
  Fine Mentre

  Scrivi("La somma è " & somma)
Fine Programma
```


Il ciclo While

(1)

- E in Visual Basic?

While CONDIZIONE
BLOCCO ISTRUZIONI
Wend

- Perché Wend e non End While?
- Microsoft ha scelto così... dubito ci sia un motivo

Il ciclo While

(1)

- E in Visual Basic?
 While CONDIZIONE
 BLOCCO ISTRUZIONI
 Wend
- Perché Wend e non End While?
- Microsoft ha scelto così... dubito ci sia un motivo

Il ciclo While

(1)

- E in Visual Basic?
 While CONDIZIONE
 BLOCCO ISTRUZIONI
 Wend
- Perché Wend e non End While?
- Microsoft ha scelto così... dubito ci sia un motivo

Il ciclo While

(1)

- E in Visual Basic?
 While CONDIZIONE
 BLOCCO ISTRUZIONI
 Wend
- Perché Wend e non End While?
- Microsoft ha scelto così... dubito ci sia un motivo

Il ciclo While

(2)

Esempio (Calcolo della somma: programma VB)

```
Sub main()  
  Dim x As Integer  
  Dim somma As Integer  
  somma = 0  
  x = InputBox("Inserisci un numero, -1 per terminare")  
  While x <> -1  
    somma = somma + x  
    x = InputBox("Inserisci un numero, -1 per terminare")  
  Wend  
  MsgBox("La somma è " & somma)  
End Sub
```

- Qual è il problema con questo programma?
- Possiamo eseguire la somma di qualsiasi sequenza?
- Se volessimo sommare la sequenza 2, -1, 3?

Il ciclo While

(2)

Esempio (Calcolo della somma: programma VB)

```
Sub main()  
    Dim x As Integer  
    Dim somma As Integer  
    somma = 0  
    x = InputBox("Inserisci un numero, -1 per terminare")  
    While x <> -1  
        somma = somma + x  
        x = InputBox("Inserisci un numero, -1 per terminare")  
    Wend  
    MsgBox("La somma è " & somma)  
End Sub
```

- Qual è il problema con questo programma?
- Possiamo eseguire la somma di qualsiasi sequenza?
- Se volessimo sommare la sequenza 2, -1, 3?

Il ciclo While

(2)

Esempio (Calcolo della somma: programma VB)

```
Sub main()  
    Dim x As Integer  
    Dim somma As Integer  
    somma = 0  
    x = InputBox("Inserisci un numero, -1 per terminare")  
    While x <> -1  
        somma = somma + x  
        x = InputBox("Inserisci un numero, -1 per terminare")  
    Wend  
    MsgBox("La somma è " & somma)  
End Sub
```

- Qual è il problema con questo programma?
- Possiamo eseguire la somma di qualsiasi sequenza?
- Se volessimo sommare la sequenza 2, -1, 3?

Il ciclo While

(2)

Esempio (Calcolo della somma: programma VB)

```
Sub main()  
    Dim x As Integer  
    Dim somma As Integer  
    somma = 0  
    x = InputBox("Inserisci un numero, -1 per terminare")  
    While x <> -1  
        somma = somma + x  
        x = InputBox("Inserisci un numero, -1 per terminare")  
    Wend  
    MsgBox("La somma è " & somma)  
End Sub
```

- Qual è il problema con questo programma?
- Possiamo eseguire la somma di qualsiasi sequenza?
- Se volessimo sommare la sequenza 2, -1, 3?

Il ciclo While

(3)

- Come possiamo aggirare questo problema?
- Leggendo prima di ogni altra cosa il numero di elementi della sequenza
- Ad esempio, l'input 3, 13, 4, 7
 - Indica che ci sono 3 elementi nella sequenza
 - La sequenza è 13, 4, 7
- **NOTA: NON** serve un numero speciale che indichi che la sequenza è finita

Il ciclo While

(3)

- Come possiamo aggirare questo problema?
- Leggendo prima di ogni altra cosa il numero di elementi della sequenza
- Ad esempio, l'input 3, 13, 4, 7
 - Indica che ci sono 3 elementi nella sequenza
 - La sequenza è 13, 4, 7
- **NOTA: NON** serve un numero speciale che indichi che la sequenza è finita

Il ciclo While

(3)

- Come possiamo aggirare questo problema?
- Leggendo prima di ogni altra cosa il numero di elementi della sequenza
- Ad esempio, l'input 3, 13, 4, 7
 - Indica che ci sono 3 elementi nella sequenza
 - La sequenza è 13, 4, 7
- **NOTA:** **NON** serve un numero speciale che indichi che la sequenza è finita

Il ciclo While

(3)

- Come possiamo aggirare questo problema?
- Leggendo prima di ogni altra cosa il numero di elementi della sequenza
- Ad esempio, l'input 3, 13, 4, 7
 - Indica che ci sono 3 elementi nella sequenza
 - La sequenza è 13, 4, 7
- **NOTA: NON** serve un numero speciale che indichi che la sequenza è finita

Il ciclo While

(4)

Esempio (Calcolo della somma: programma VB - versione 2)

```
Sub main()  
    Dim n As Integer  
    Dim x As Integer  
    Dim i As Integer  
    Dim somma As Integer  
  
    somma = 0  
  
    n = InputBox("Quanti numeri nella sequenza?")  
  
    i = 1  
    While i <= n  
        x = InputBox("Inserisci un numero")  
        somma = somma + x  
        i = i + 1  
    Wend  
  
    MsgBox("La somma è " & somma)  
End Sub
```

Il ciclo For

(1)

- In Visual Basic c'è un altro tipo di ciclo
 - For **VARIABILE = VALORE_INIZIALE To VALORE_FINALE**
BLOCCO ISTRUZIONI
 - Next **VARIABILE**
- VARIABILE deve essere una variabile intera
- VARIABILE viene settata a VALORE_INIZIALE
- BLOCCO ISTRUZIONI viene eseguito
VARIABILE \leq VALORE_FINALE
- Next VARIABILE incrementa VARIABILE e riesegue il ciclo

Il ciclo For

(1)

- In Visual Basic c'è un altro tipo di ciclo
 - For **VARIABILE = VALORE_INIZIALE To VALORE_FINALE**
BLOCCO ISTRUZIONI
 - Next **VARIABILE**
- VARIABILE deve essere una variabile intera
- VARIABILE viene settata a VALORE_INIZIALE
- BLOCCO ISTRUZIONI viene eseguito
VARIABILE <= VALORE_FINALE
- Next VARIABILE incrementa VARIABILE e riesegue il ciclo

Il ciclo For

(1)

- In Visual Basic c'è un altro tipo di ciclo
 - For **VARIABILE = VALORE_INIZIALE To VALORE_FINALE**
BLOCCO ISTRUZIONI
 - Next **VARIABILE**
- VARIABILE deve essere una variabile intera
- VARIABILE viene settata a VALORE_INIZIALE
- BLOCCO ISTRUZIONI viene eseguito
VARIABILE \leq VALORE_FINALE
- Next VARIABILE incrementa VARIABILE e riesegue il ciclo

Il ciclo For

(1)

- In Visual Basic c'è un altro tipo di ciclo
 - For **VARIABILE** = VALORE_INIZIALE To **VALORE_FINALE**
BLOCCO ISTRUZIONI
 - Next **VARIABILE**
- **VARIABILE** deve essere una variabile intera
- **VARIABILE** viene settata a **VALORE_INIZIALE**
- **BLOCCO ISTRUZIONI** viene eseguito
VARIABILE <= **VALORE_FINALE**
- Next **VARIABILE** incrementa **VARIABILE** e riesegue il ciclo

Il ciclo For

(2)

Esempio (Calcolo della somma: programma VB - versione 3)

```
Sub main()  
    Dim n As Integer  
    Dim x As Integer  
    Dim i As Integer  
    Dim somma As Integer  
  
    somma = 0  
  
    n = InputBox("Quanti numeri nella sequenza?")  
  
    For i = 1 To n  
        x = InputBox("Inserisci un numero")  
        somma = somma + x  
    Next i  
  
    MsgBox("La somma è " & somma)  
End Sub
```

While vs. For

- Usare il While o il For fa poca differenza
- Potete usare quello che più vi piace
- Per alcuni tipi di ciclo (come quello appena visto), il For è più comodo
- Ma il While è più flessibile
- Provate a leggere una sequenza terminata da -1 con un For
 - Non si può fare... a meno di trucchetti!

While vs. For

- Usare il While o il For fa poca differenza
- Potete usare quello che più vi piace
- Per alcuni tipi di ciclo (come quello appena visto), il For è più comodo
- Ma il While è più flessibile
- Provate a leggere una sequenza terminata da -1 con un For
 - Non si può fare... a meno di trucchetti!

While vs. For

- Usare il While o il For fa poca differenza
- Potete usare quello che più vi piace
- Per alcuni tipi di ciclo (come quello appena visto), il For è più comodo
- Ma il While è più flessibile
- Provate a leggere una sequenza terminata da -1 con un For
 - Non si può fare... a meno di trucchetti!

While vs. For

- Usare il While o il For fa poca differenza
- Potete usare quello che più vi piace
- Per alcuni tipi di ciclo (come quello appena visto), il For è più comodo
- Ma il While è più flessibile
- Provate a leggere una sequenza terminata da -1 con un For
 - Non si può fare... a meno di trucchetti!

While vs. For

- Usare il While o il For fa poca differenza
- Potete usare quello che più vi piace
- Per alcuni tipi di ciclo (come quello appena visto), il For è più comodo
- Ma il While è più flessibile
- Provate a leggere una sequenza terminata da -1 con un For
 - Non si può fare... a meno di trucchetti!

Calcolo della media

(1)

- La nostra intenzione era di scrivere un programma per il calcolo della media
- Siamo interessati, quindi, a rappresentare la formula

$$m = \frac{\sum_{i=1}^n x_i}{n}$$

- Abbiamo due possibilità
 - 1 Inserire una sequenza terminata da -1
 - 2 Inserire la lunghezza prima della sequenza

Calcolo della media

(1)

- La nostra intenzione era di scrivere un programma per il calcolo della media
- Siamo interessati, quindi, a rappresentare la formula

$$m = \frac{\sum_{i=1}^n x_i}{n}$$

- Abbiamo due possibilità
 - 1 Inserire una sequenza terminata da -1
 - 2 Inserire la lunghezza prima della sequenza

Calcolo della media

(1)

- La nostra intenzione era di scrivere un programma per il calcolo della media
- Siamo interessati, quindi, a rappresentare la formula

$$m = \frac{\sum_{i=1}^n x_i}{n}$$

- Abbiamo due possibilità
 - 1 Inserire una sequenza terminata da -1
 - 2 Inserire la lunghezza prima della sequenza

Calcolo della media

(1)

- La nostra intenzione era di scrivere un programma per il calcolo della media
- Siamo interessati, quindi, a rappresentare la formula

$$m = \frac{\sum_{i=1}^n x_i}{n}$$

- Abbiamo due possibilità
 - 1 Inserire una sequenza terminata da -1
 - 2 Inserire la lunghezza prima della sequenza

Calcolo della media

(1)

- La nostra intenzione era di scrivere un programma per il calcolo della media
- Siamo interessati, quindi, a rappresentare la formula

$$m = \frac{\sum_{i=1}^n x_i}{n}$$

- Abbiamo due possibilità
 - 1 Inserire una sequenza terminata da -1
 - 2 Inserire la lunghezza prima della sequenza

Calcolo della media

(2)

- 1 Inserire una sequenza terminata da -1

Esempio (Calcolo della media)

```
Sub main()  
  Dim x As Integer  
  Dim somma As Integer  
  Dim contatore As Integer  
  somma = 0  
  contatore = 0  
  x = InputBox("Inserisci un numero, -1 per terminare")  
  While x <> -1  
    somma = somma + x  
    contatore = contatore + 1  
    x = InputBox("Inserisci un numero, -1 per terminare")  
  Wend  
  MsgBox("La media è " & somma / contatore)  
End Sub
```

- **ATTENZIONE:** I programmi possono contenere errori
 - Sintattici, segnalati da Visual Basic
 - Semantici, più difficili da trovare

Calcolo della media

(2)

- 1 Inserire una sequenza terminata da -1

Esempio (Calcolo della media)

```
Sub main()  
  Dim x As Integer  
  Dim somma As Integer  
  Dim contatore As Integer  
  somma = 0  
  contatore = 0  
  x = InputBox("Inserisci un numero, -1 per terminare")  
  While x <> -1  
    somma = somma + x  
    contatore = contatore + 1  
    x = InputBox("Inserisci un numero, -1 per terminare")  
  Wend  
  MsgBox("La media è " & somma / contatore)  
End Sub
```

- **ATTENZIONE:** I programmi possono contenere errori
 - Sintattici, segnalati da Visual Basic
 - Semantici, più difficili da trovare

Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```

Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```


Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```

Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```

Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```

Anomalie: errori semantici

(1)

- Il programma appena scritto contiene un errore semantico
- Provate a inserire la sequenza vuota: -1
- Il programma esegue una divisione per 0
 - Non è definita
- Del resto, la media di 0 valori non è definita
- Correzione

```
If contatore <> 0 Then
    MsgBox("La media è " & somma / contatore)
Else
    MsgBox("Media indefinita")
End If
```

Anomalie: errori semantici

(2)

Definizione (Errore semantico)

Gli errori di semantica sono comportamenti non voluti o inattesi del programma.

- Nel caso di una divisione per 0 è chiara la presenza dell'errore
- Purtroppo non è sempre così
- Un errore di semantica può essere anche una formula sbagliata
- Ad esempio, se nel programma della somma sostituiamo + con -
$$\text{somma} = \text{somma} - x$$

Anomalie: errori semantici

(2)

Definizione (Errore semantico)

Gli errori di semantica sono comportamenti non voluti o inattesi del programma.

- Nel caso di una divisione per 0 è chiara la presenza dell'errore
- Purtroppo non è sempre così
- Un errore di semantica può essere anche una formula sbagliata
- Ad esempio, se nel programma della somma sostituiamo + con -
$$\text{somma} = \text{somma} - x$$

Anomalie: errori semantici

(2)

Definizione (Errore semantico)

Gli errori di semantica sono comportamenti non voluti o inattesi del programma.

- Nel caso di una divisione per 0 è chiara la presenza dell'errore
- Purtroppo non è sempre così
- Un errore di semantica può essere anche una formula sbagliata
- Ad esempio, se nel programma della somma sostituiamo + con -

somma = somma - x

Anomalie: errori semantici

(2)

Definizione (Errore semantico)

Gli errori di semantica sono comportamenti non voluti o inattesi del programma.

- Nel caso di una divisione per 0 è chiara la presenza dell'errore
- Purtroppo non è sempre così
- Un errore di semantica può essere anche una formula sbagliata
- Ad esempio, se nel programma della somma sostituiamo + con -

somma = somma - x

Anomalie: errori semantici

(2)

Definizione (Errore semantico)

Gli errori di semantica sono comportamenti non voluti o inattesi del programma.

- Nel caso di una divisione per 0 è chiara la presenza dell'errore
- Purtroppo non è sempre così
- Un errore di semantica può essere anche una formula sbagliata
- Ad esempio, se nel programma della somma sostituiamo + con -

somma = somma - x

Calcolo del prodotto

(1)

Esempio (Prodotto di 5 interi)

Calcolare il prodotto di 5 numeri interi.

```
Sub main()  
  Dim x As Integer  
  Dim i As Integer  
  Dim prodotto As Integer  
  
  prodotto = 0  
  For i = 1 To 5  
    x = InputBox("Inserisci un numero")  
    prodotto = prodotto * x  
  Next i  
  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **ATTENZIONE:** c'è un errore semantico
- Il programma stamperà sempre 0
- “prodotto” deve essere inizializzato all'elemento neutro: 1

Calcolo del prodotto

(1)

Esempio (Prodotto di 5 interi)

Calcolare il prodotto di 5 numeri interi.

```
Sub main()  
  Dim x As Integer  
  Dim i As Integer  
  Dim prodotto As Integer  
  
  prodotto = 0  
  For i = 1 To 5  
    x = InputBox("Inserisci un numero")  
    prodotto = prodotto * x  
  Next i  
  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **ATTENZIONE:** c'è un errore semantico
- Il programma stamperà sempre 0
- “prodotto” deve essere inizializzato all'elemento neutro: 1

Calcolo del prodotto

(1)

Esempio (Prodotto di 5 interi)

Calcolare il prodotto di 5 numeri interi.

```
Sub main()  
  Dim x As Integer  
  Dim i As Integer  
  Dim prodotto As Integer  
  
  prodotto = 0  
  For i = 1 To 5  
    x = InputBox("Inserisci un numero")  
    prodotto = prodotto * x  
  Next i  
  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **ATTENZIONE:** c'è un errore semantico
- Il programma stamperà sempre 0
- “prodotto” deve essere inizializzato all'elemento neutro: 1

Calcolo del prodotto

(1)

Esempio (Prodotto di 5 interi)

Calcolare il prodotto di 5 numeri interi.

```
Sub main()  
  Dim x As Integer  
  Dim i As Integer  
  Dim prodotto As Integer  
  
  prodotto = 0  
  For i = 1 To 5  
    x = InputBox("Inserisci un numero")  
    prodotto = prodotto * x  
  Next i  
  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **ATTENZIONE:** c'è un errore semantico
- Il programma stamperà sempre 0
- “prodotto” deve essere inizializzato all'elemento neutro: 1

Calcolo del prodotto

(2)

Esempio (Prodotto di n interi)

Calcolare il prodotto di una sequenza di n numeri interi.

```
Sub main()  
  Dim n As Integer  
  Dim x As Integer  
  Dim i As Integer  
  Dim prodotto As Integer  
  
  n = InputBox("Quanti numeri devo moltiplicare?")  
  
  prodotto = 1  
  For i = 1 To n  
    x = InputBox("Inserisci un numero")  
    prodotto = prodotto * x  
  Next i  
  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

Calcolo del prodotto

(3)

Esempio (Prodotto di interi positivi terminati da -1)

Calcolare il prodotto di una sequenza di numeri interi positivi, terminata da -1.

```
Sub main()  
    Dim x As Integer  
    Dim prodotto As Integer  
    prodotto = 1  
    x = InputBox("Inserisci un numero, -1 per terminare")  
    While x <> -1  
        prodotto = prodotto * x  
        x = InputBox("Inserisci un numero, -1 per terminare")  
    Wend  
    MsgBox("Il prodotto è " & prodotto)  
End Sub
```

Calcolo del prodotto

(3)

Esempio (Prodotto di interi positivi terminati da -1)

Calcolare il prodotto di una sequenza di numeri interi positivi, terminata da -1.

```
Sub main()  
    Dim x As Integer  
    Dim prodotto As Integer  
    prodotto = 1  
    x = InputBox("Inserisci un numero, -1 per terminare")  
    While x <> -1  
        prodotto = prodotto * x  
        x = InputBox("Inserisci un numero, -1 per terminare")  
    Wend  
    MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **ATTENZIONE:** c'è un errore semantico
- Cosa succede se l'input è: 2, -2, -1? L'output è -4
- Ma -2 non è un intero positivo

Calcolo del prodotto

(3)

Esempio (Prodotto di interi positivi terminati da -1)

Calcolare il prodotto di una sequenza di numeri interi positivi, terminata da -1.

```
Sub main()  
  Dim x As Integer  
  Dim prodotto As Integer  
  prodotto = 1  
  x = InputBox("Inserisci un numero, -1 per terminare")  
  While x <> -1  
    prodotto = prodotto * x  
    x = InputBox("Inserisci un numero, -1 per terminare")  
  Wend  
  MsgBox("Il prodotto è " & prodotto)  
End Sub
```

- **CORREZIONE:** All'interno del While

```
  If x > 0 Then  
    prodotto = prodotto * x  
  Else  
    MsgBox("Numero scartato " & x)  
  End If
```

Esercizi

- 1 Data una sequenza di 5 interi, calcolare separatamente la somma dei numeri dispari e la somma dei numeri pari
 - Ad esempio, per 1, 2, 3, 4, 5 avremo
 - $\text{somma_pari} = 2 + 4 = 6$
 - $\text{somma_dispari} = 1 + 3 + 5 = 9$
- 2 Generalizzare l'esercizio 1 a una sequenza di n interi
- 3 Generalizzare l'esercizio 1 a una sequenza di interi positivi terminata da -1
- 4 Data una sequenza di 7 temperature, determinare quante sono ≥ 40 e quante ≤ 4
- 5 Generalizzare l'esercizio 4 a una sequenza di n temperature
- 6 Generalizzare l'esercizio 4 a una sequenza di temperature maggiori di -30
 - Scegliere opportunamente il numero per terminare la sequenza