

Dal problema al programma: le basi della programmazione

L'istruzione di iterazione (o ciclo)

prof. Leonardo Essam Dei Rossi

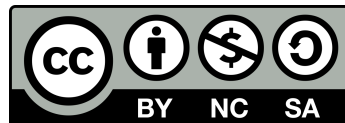
ITT "M. Buonarroti" - Trento (TN)

Anno scolastico 2025/2026

Questo materiale è disponibile sul sito Web del docente per il corso di [Tecnologie informatiche](#) per le studentesse e gli studenti dell'anno scolastico 2025/2026.

Versione: 1.3.3 (B)
Ultima modifica: 16/03/2026 18:42
Riferimenti: [1, Cap. 7 - Lez. 5]

This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](#)



- 1 L'istruzione di iterazione (o ciclo)
 - Esempi di istruzione di iterazione
 - Codifica dell'iterazione
- 2 L'iterazione indefinita
 - Esempio #1: Mi compro la moto!
 - Il tracciamento delle variabili
- 3 L'iterazione definita
 - Esempio #2: Potenza n-esima di un numero
 - Esercizio #1: Pseudocodifica di Esempio #2 (1)
 - Esercizio #2: Pseudocodifica di Esempio #2 (2)

- 1 L'istruzione di iterazione (o ciclo)
 - Esempi di istruzione di iterazione
 - Codifica dell'iterazione

- 2 L'iterazione indefinita
 - Esempio #1: Mi compro la moto!
 - Il tracciamento delle variabili

- 3 L'iterazione definita
 - Esempio #2: Potenza n-esima di un numero
 - Esercizio #1: Pseudocodifica di Esempio #2 (1)
 - Esercizio #2: Pseudocodifica di Esempio #2 (2)

L'istruzione di iterazione (o ciclo)

Nella descrizione di un **algoritmo** può capitare che alcune operazioni debbano essere eseguite più di una volta, cioè **ripetute** in modo identico.

Se dobbiamo effettuare il conto alla rovescia da 10 a 1, dobbiamo ripetere per 10 volte l'operazione "sottrai 1", mentre se dobbiamo riempire d'acqua una vasca da bagno utilizzando un bicchiere è necessario versare ripetutamente il contenuto del bicchiere finché la vasca non risulti piena.

L'istruzione di iterazione (o ciclo)

Nella descrizione di un **algoritmo** può capitare che alcune operazioni debbano essere eseguite più di una volta, cioè **ripetute** in modo identico.

Se dobbiamo effettuare il conto alla rovescia da 10 a 1, dobbiamo ripetere per 10 volte l'operazione "sottrai 1", mentre se dobbiamo riempire d'acqua una vasca da bagno utilizzando un bicchiere è necessario versare ripetutamente il contenuto del bicchiere finché la vasca non risulti piena.

L'istruzione di iterazione (o ciclo)

La ripetizione di un insieme di istruzioni prende il nome di **iterazione** o ciclo (in inglese: *loop*); con questa istruzione indichiamo al calcolatore di **ripetere un gruppo di istruzioni**:

- Un numero di volte ben **determinato**, come nel primo esempio;
- **Finché** non si **verifica** una situazione, come nel secondo esempio.

Definizione: Iterazione

Struttura di controllo che permette di eseguire più volte un'istruzione. Nella maggior parte dei casi occorre indicare, oltre all'istruzione che dovrà essere iterata, anche la condizione per uscire dall'iterazione stessa.

L'istruzione di iterazione (o ciclo)

La ripetizione di un insieme di istruzioni prende il nome di **iterazione** o ciclo (in inglese: *loop*); con questa istruzione indichiamo al calcolatore di **ripetere un gruppo di istruzioni**:

- Un numero di volte ben **determinato**, come nel primo esempio;
- **Finché** non si **verifica** una situazione, come nel secondo esempio.

Definizione: Iterazione

Struttura di controllo che permette di eseguire più volte un'istruzione. Nella maggior parte dei casi occorre indicare, oltre all'istruzione che dovrà essere iterata, anche la condizione per uscire dall'iterazione stessa.

L'istruzione di iterazione (o ciclo)

La ripetizione di un insieme di istruzioni prende il nome di **iterazione** o ciclo (in inglese: *loop*); con questa istruzione indichiamo al calcolatore di **ripetere un gruppo di istruzioni**:

- Un numero di volte ben **determinato**, come nel primo esempio;
- **Finché** non si **verifica** una situazione, come nel secondo esempio.

Definizione: Iterazione

Struttura di controllo che permette di eseguire più volte un'istruzione. Nella maggior parte dei casi occorre indicare, oltre all'istruzione che dovrà essere iterata, anche la condizione per uscire dall'iterazione stessa.

L'istruzione di iterazione (o ciclo)

La ripetizione di un insieme di istruzioni prende il nome di **iterazione** o ciclo (in inglese: *loop*); con questa istruzione indichiamo al calcolatore di **ripetere un gruppo di istruzioni**:

- Un numero di volte ben **determinato**, come nel primo esempio;
- **Finché** non si **verifica** una situazione, come nel secondo esempio.

Definizione: Iterazione

Struttura di controllo che permette di **eseguire più volte** un'istruzione. Nella maggior parte dei casi occorre indicare, oltre all'istruzione che dovrà essere iterata, anche la condizione per uscire dall'iterazione stessa.

L'istruzione di iterazione (o ciclo)

Il gruppo di **istruzioni** ripetute prende il nome di **corpo del ciclo**.

L'istruzione di **iterazione** viene classificata proprio per la caratteristica di conoscere o meno a priori il **numero di ripetizioni** (del corpo del ciclo) che deve effettuare:

Quando il numero di ripetizioni è noto a priori, cioè sai quante volte le istruzioni devono essere ripetute, il ciclo prende il nome di **iterazione definita**.

Quando invece il ciclo viene ripetuto un numero di volte sconosciuto a priori e termina quando si verifica una particolare condizione, l'iterazione si dice **indefinita**.

L'istruzione di iterazione (o ciclo)

Il gruppo di **istruzioni** ripetute prende il nome di **corpo del ciclo**.

L'istruzione di **iterazione** viene classificata proprio per la caratteristica di conoscere o meno a priori il **numero di ripetizioni** (del corpo del ciclo) che deve effettuare:

Quando il numero di ripetizioni è noto a priori, cioè sai quante volte le istruzioni devono essere ripetute, il ciclo prende il nome di **iterazione definita**.

Quando invece il ciclo viene ripetuto un numero di volte sconosciuto a priori e termina quando si verifica una particolare condizione, l'iterazione si dice **indefinita**.

L'istruzione di iterazione (o ciclo)

Il gruppo di **istruzioni** ripetute prende il nome di **corpo del ciclo**.

L'istruzione di **iterazione** viene classificata proprio per la caratteristica di conoscere o meno a priori il **numero di ripetizioni** (del corpo del ciclo) che deve effettuare:

Quando il numero di ripetizioni è noto a priori, cioè sai quante volte le istruzioni devono essere ripetute, il ciclo prende il nome di **iterazione definita**.

Quando invece il ciclo viene ripetuto un numero di volte sconosciuto a priori e termina quando si verifica una particolare condizione, l'iterazione si dice **indefinita**.

L'istruzione di iterazione (o ciclo)

Esempi di istruzione di iterazione

Vediamo alcuni esempi dei due diversi tipi di **istruzione di iterazione**.

Iterazioni definite	Iterazioni indefinite
Fai 10 giorni di ferie	Mescola la pasta finché è cotta
Leggi 20 pagine	Innaffia le piante finché basta
Per 10 volte scrivi “devo studiare di più”	Mangia finché sei sazio
Attacca 30 figurine sull'album	Mentre piove fatti una dormita
Invia 100 sms alla zia	Mentre non sei stanco, studia
Imbianca con 3 mani la parete	Finché c'è vita c'è speranza
Tra 85 giorni è finita la scuola	Mentre il serbatoio non è pieno metti benzina
Per 210 giorni devi andare a scuola	Dormi finché hai sonno

L'istruzione di iterazione (o ciclo)

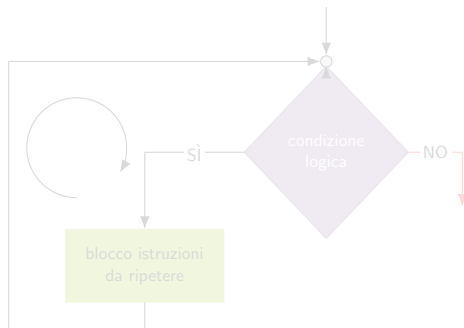
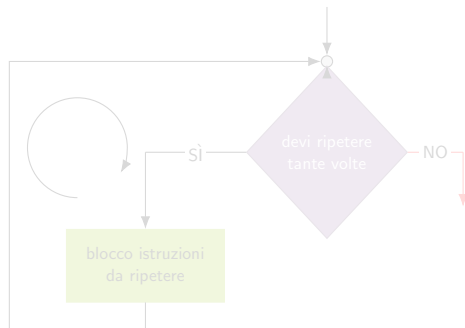
Esempi di istruzione di iterazione

Osservando attentamente i due gruppi di istruzioni possiamo affermare che nelle **iterazioni definite** è sempre presente un **numero**, cioè viene sempre indicato chiaramente quante volte deve essere eseguita un'operazione, mentre in quelle **indefinite** non si sa a priori per quante volte questa debba essere ripetuta.

L'istruzione di iterazione (o ciclo)

Codifica dell'iterazione

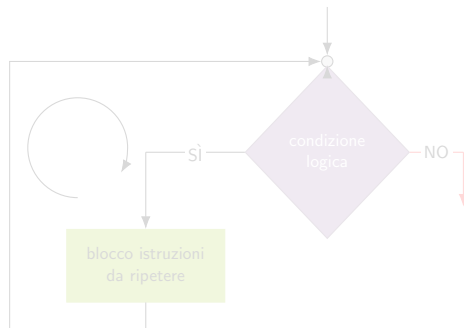
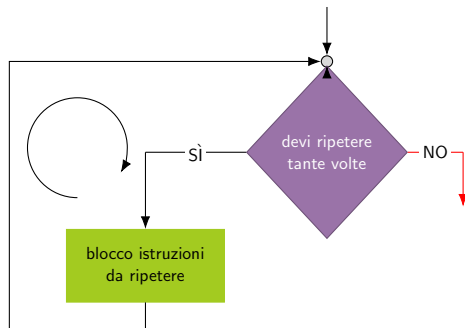
Lo schema generale di un'istruzione **iterativa** è riportato nelle figure seguenti, dove possiamo osservare come il percorso di un ramo formi un “**anello**” con l'istruzione di **test**.



L'istruzione di iterazione (o ciclo)

Codifica dell'iterazione

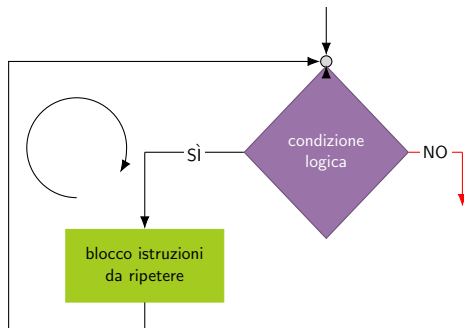
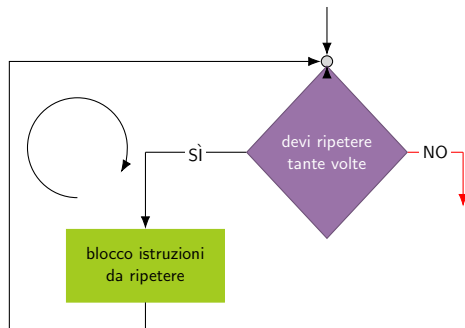
Lo schema generale di un'istruzione **iterativa** è riportato nelle figure seguenti, dove possiamo osservare come il percorso di un ramo formi un “**anello**” con l'istruzione di **test**.



L'istruzione di iterazione (o ciclo)

Codifica dell'iterazione

Lo schema generale di un'istruzione **iterativa** è riportato nelle figure seguenti, dove possiamo osservare come il percorso di un ramo formi un “**anello**” con l'istruzione di **test**.



L'istruzione di iterazione (o ciclo)

Codifica dell'iterazione

Il **test** controlla se deve essere **ripetuto** il blocco di istruzioni, oppure se la **ripetizione** del **ciclo** termina e quindi si deve “uscire” dalla parte di destra dove prosegue il programma.

Possiamo osservare che gli schemi sono identici, cambia solamente il tipo di **test**:

- Nel primo caso il controllo viene fatto sul valore del **contatore** che contiene il numero di volte che il **ciclo** è stato eseguito confrontandolo con il valore totale predefinito;
- Nel secondo caso come **test** che ci “fa entrare in loop” abbiamo una qualunque **condizione logica**.

L'istruzione di iterazione (o ciclo)

Codifica dell'iterazione

Il **test** controlla se deve essere **ripetuto** il blocco di istruzioni, oppure se la **ripetizione** del **ciclo** termina e quindi si deve “uscire” dalla parte di destra dove prosegue il programma.

Possiamo osservare che gli schemi sono identici, cambia solamente il tipo di **test**:

- Nel primo caso il controllo viene fatto sul valore del **contatore** che contiene il numero di volte che il **ciclo** è stato eseguito confrontandolo con il valore totale predefinito;
- Nel secondo caso come **test** che ci “fa entrare in loop” abbiamo una qualunque **condizione logica**.

- 1 L'istruzione di iterazione (o ciclo)
 - Esempi di istruzione di iterazione
 - Codifica dell'iterazione

- 2 L'iterazione indefinita
 - Esempio #1: Mi compro la moto!
 - Il tracciamento delle variabili

- 3 L'iterazione definita
 - Esempio #2: Potenza n-esima di un numero
 - Esercizio #1: Pseudocodifica di Esempio #2 (1)
 - Esercizio #2: Pseudocodifica di Esempio #2 (2)

L'iterazione indefinita

Esistono due diverse istruzioni di **iterazione indefinita** ma dato che sono equivalenti in questa trattazione descriveremo solo quella più utilizzata, e cioè quella che prende il nome di **iterazione precondizionata** detta anche “**ciclo a condizione iniziale**” o “**a controllo di testa**”.

In tutti e tre i nomi sopra riportati possiamo sottolineare come venga indicato il momento in cui viene fatto il **test**, cioè quando viene eseguito il controllo che permette di stabilire se bisogna entrare nel **ramo di sinistra** ed eseguire (o **ripetere**) le istruzioni del **ciclo** oppure si esce dal **ramo di destra** e si prosegue con il resto del programma.

I programmatori la chiamano “**ciclo while**” facendo riferimento alla parola riservata con la quale viene codificata.

L'iterazione indefinita

Esistono due diverse istruzioni di **iterazione indefinita** ma dato che sono equivalenti in questa trattazione descriveremo solo quella più utilizzata, e cioè quella che prende il nome di **iterazione precondizionata** detta anche “**ciclo a condizione iniziale**” o “**a controllo di testa**”.

In tutti e tre i nomi sopra riportati possiamo sottolineare come venga indicato il momento in cui viene fatto il **test**, cioè quando viene eseguito il controllo che permette di stabilire se bisogna entrare nel **ramo di sinistra** ed eseguire (o **ripetere**) le istruzioni del **ciclo** oppure si esce dal **ramo di destra** e si prosegue con il resto del programma.

I programmatori la chiamano “**ciclo while**” facendo riferimento alla parola riservata con la quale viene codificata.

L'iterazione indefinita

Esistono due diverse istruzioni di **iterazione indefinita** ma dato che sono equivalenti in questa trattazione descriveremo solo quella più utilizzata, e cioè quella che prende il nome di **iterazione precondizionata** detta anche “**ciclo a condizione iniziale**” o “**a controllo di testa**”.

In tutti e tre i nomi sopra riportati possiamo sottolineare come venga indicato il momento in cui viene fatto il **test**, cioè quando viene eseguito il controllo che permette di stabilire se bisogna entrare nel **ramo di sinistra** ed eseguire (o **ripetere**) le istruzioni del **ciclo** oppure si esce dal **ramo di destra** e si prosegue con il resto del programma.

I programmatori la chiamano “**ciclo while**” facendo riferimento alla parola riservata con la quale viene codificata.

L'iterazione indefinita

Esempio #1: Mi compro la moto!

Vediamo un primo programma dove viene chiesto a un utente di inserire alcuni numeri fino a che non si raggiunga un valore prefissato: supponiamo, ad esempio, che per comperare la moto servano 800 € e quando con i nostri risparmi raggiungiamo tale cifra il programma ci avvisa mediante un messaggio.

In questo programma abbiamo bisogno di due **variabili** rispettivamente per memorizzare il costo della moto e "l'ammontare dei nostri risparmi": le chiameremo **costoMoto** e **risparmi**.

L'iterazione indefinita

Esempio #1: Mi compro la moto!

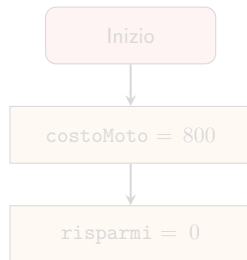
Vediamo un primo programma dove viene chiesto a un utente di inserire alcuni numeri fino a che non si raggiunga un valore prefissato: supponiamo, ad esempio, che per comperare la moto servano 800 € e quando con i nostri risparmi raggiungiamo tale cifra il programma ci avvisa mediante un messaggio.

In questo programma abbiamo bisogno di due **variabili** rispettivamente per memorizzare il costo della moto e “l'ammontare dei nostri risparmi”: le chiameremo **costoMoto** e **risparmi**.

L'iterazione indefinita

Esempio #1: Mi compro la moto!

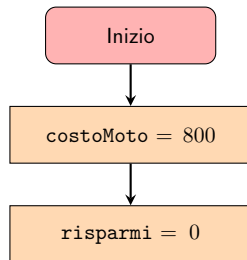
Il programma deve ricevere con **INPUT** le nuove somme che risparmiamo e aggiungerle al totale **risparmi**: a ogni nuovo valore che assume questa **variabile** è necessario eseguire il confronto con **costoMoto** per verificare se tale importo è stato raggiunto e, di conseguenza, è possibile effettuare il nostro acquisto.



L'iterazione indefinita

Esempio #1: Mi compro la moto!

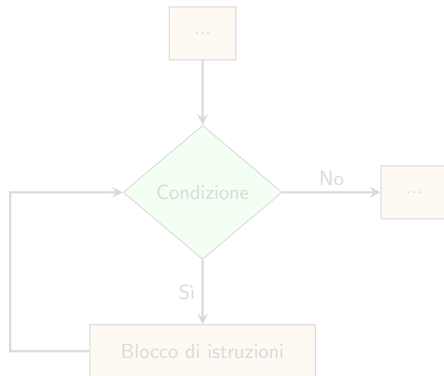
Il programma deve ricevere con **INPUT** le nuove somme che risparmiamo e aggiungerle al totale **risparmi**: a ogni nuovo valore che assume questa **variabile** è necessario eseguire il confronto con **costoMoto** per verificare se tale importo è stato raggiunto e, di conseguenza, è possibile effettuare il nostro acquisto.



L'iterazione indefinita

Esempio #1: Mi compro la moto!

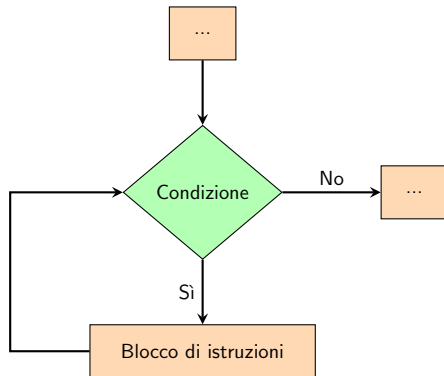
Inseriamo ora l'istruzione di **iterazione indefinita**, che permettere di ripetere più volte un ramo del diagramma:



L'iterazione indefinita

Esempio #1: Mi compro la moto!

Inseriamo ora l'istruzione di **iterazione indefinita**, che permettere di ripetere più volte un ramo del diagramma:



L'iterazione indefinita

Esempio #1: Mi compro la moto!

L'istruzione che controlla l'**ingresso**
dell'istruzione di **iterazione** è la seguente:

```
(risparmi < costoMoto)
```

Questo test è l'istruzione che determina se dobbiamo “ripetere il ciclo” nel caso di esito **VERO (T)**, oppure se possiamo proseguire con il resto del programma, caso **FALSO (F)**: è un **test** come quelli precedentemente visti per l'**istruzione di selezione**, anche se graicamente, per cercare di “ridurre l'ingombro”, il ramo **T** viene “fatto uscire” dal basso e non dallo spigolo di sinistra del rombo.

L'iterazione indefinita

Esempio #1: Mi compro la moto!

L'istruzione che controlla l'**ingresso** dell'istruzione di **iterazione** è la seguente:

```
(risparmi < costoMoto)
```

Questo test è l'istruzione che determina se dobbiamo “ripetere il ciclo” nel caso di esito **VERO (T)**, oppure se possiamo proseguire con il resto del programma, caso **FALSO (F)**: è un **test** come quelli precedentemente visti per l'**istruzione di selezione**, anche se graicamente, per cercare di “ridurre l'ingombro”, il ramo **T** viene “fatto uscire” dal basso e non dallo spigolo di sinistra del rombo.

L'iterazione indefinita

Esempio #1: Mi compro la moto!

Le istruzioni che vengono ripetute
“periodicamente” sono riportate a fianco.

Abbiamo aggiunto la variabile **nuovo** che
contiene l'importo del nuovo risparmio da
aggiungere al nostro “tesoretto”.

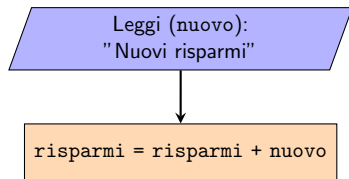


L'iterazione indefinita

Esempio #1: Mi compro la moto!

Le istruzioni che vengono ripetute
“periodicamente” sono riportate a fianco.

Abbiamo aggiunto la variabile **nuovo** che
contiene l'importo del nuovo risparmio da
aggiungere al nostro “tesoretto”.



L'iterazione indefinita

Esempio #1: Mi compro la moto!

In pseudocodifica formale possiamo tradurre l'algoritmo:

```
Inizio
Calcola: (costoMoto = 800)
Calcola: (risparmi = 0)

Mentre (risparmi < costoMoto), esegui:
- Leggi (nuovo): "Inserisci nuovi risparmi"
- Calcola: (risparmi = risparmi + nuovo)

Stampa (): "Ora puoi comprare la moto!"

Fine
```

L'iterazione indefinita

Esempio #1: Mi compro la moto!

Potete testare la pseudocodifica formale proposta tramite un compilatore on-line sul sito Web del docente: [\(link\)](#)

L'iterazione indefinita

Il tracciamento delle variabili

Il tracciamento delle variabili permette di visualizzare l'esecuzione di ogni singola istruzione compresa la gestione della memoria, cioè i valori che le variabili assumono man mano che il programma evolve.

Visualizzare la traccia del programma è di fondamentale importanza per ricercare eventuali errori presenti nel codice: è uno strumento fondamentale per il **debugging**.

È comodo riportare anche la colonna che rappresenta l'istruzione di **test**, e quindi indicare il valore che assume la **variabile booleana** che ne è il risultato.

L'iterazione indefinita

Il tracciamento delle variabili

Il tracciamento delle variabili permette di visualizzare l'esecuzione di ogni singola istruzione compresa la gestione della memoria, cioè i valori che le variabili assumono man mano che il programma evolve.

Visualizzare la traccia del programma è di fondamentale importanza per ricercare eventuali errori presenti nel codice: è uno strumento fondamentale per il **debugging**.

È comodo riportare anche la colonna che rappresenta l'istruzione di **test**, e quindi indicare il valore che assume la **variabile booleana** che ne è il risultato.

L'iterazione indefinita

Il tracciamento delle variabili

Il tracciamento delle variabili permette di visualizzare l'esecuzione di ogni singola istruzione compresa la gestione della memoria, cioè i valori che le variabili assumono man mano che il programma evolve.

Visualizzare la traccia del programma è di fondamentale importanza per ricercare eventuali errori presenti nel codice: è uno strumento fondamentale per il **debugging**.

È comodo riportare anche la colonna che rappresenta l'istruzione di **test**, e quindi indicare il valore che assume la **variabile booleana** che ne è il risultato.

L'iterazione indefinita

Il tracciamento delle variabili

Consideriamo **Esempio #1: Mi compro la moto!**.

Realizziamo la **tabella di traccia del programma** (in inglese: *trace table*) di una possibile esecuzione:

Istruzioni:	risparmi < costoMoto	costoMoto	nuovo	risparmi
Dopo le inizializzazioni		800	0	0
Test	$(0 < 800) \Rightarrow \text{VERO (T)}$			
Prima ripetizione del ciclo			200	
				200
Test	$(200 < 800) \Rightarrow \text{VERO (T)}$			
Seconda ripetizione del ciclo			300	

L'iterazione indefinita

Il tracciamento delle variabili

Consideriamo **Esempio #1: Mi compro la moto!**.

Realizziamo la **tabella di traccia del programma** (in inglese: *trace table*) di una possibile esecuzione:

Istruzioni:	risparmi < costoMoto	costoMoto	nuovo	risparmi
				500
Test	$(500 < 800) \Rightarrow \text{VERO (T)}$			
Terza ripetizione del ciclo			400	
				900
Test	$(900 < 800) \Rightarrow \text{FALSO (F)}$			
Fine Ripetizione				

L'iterazione indefinita

Il tracciamento delle variabili

Possiamo notare come il valore di `costoMoto` non viene mai modificato: di fatto è una **costante** che viene **inizializzata** e successivamente utilizzata solo per i confronti.

Osserviamo come l'ultima colonna ci riporta la “cronologia” dei nostri risparmi, tenendo la “traccia della loro storia”.

Il programma funziona anche se introduciamo importi negativi, cioè se oltre che ad aggiungere nuovi risparmi parte di quelli presenti vengono prelevati e... spesi!

L'iterazione indefinita

Il tracciamento delle variabili

Possiamo notare come il valore di `costoMoto` non viene mai modificato: di fatto è una **costante** che viene **inizializzata** e successivamente utilizzata solo per i confronti.

Osserviamo come l'ultima colonna ci riporta la “cronologia” dei nostri risparmi, tenendo la “traccia della loro storia”.

Il programma funziona anche se introduciamo importi negativi, cioè se oltre che ad aggiungere nuovi risparmi parte di quelli presenti vengono prelevati e... spesi!

L'iterazione indefinita

Il tracciamento delle variabili

Possiamo notare come il valore di `costoMoto` non viene mai modificato: di fatto è una **costante** che viene **inizializzata** e successivamente utilizzata solo per i confronti.

Osserviamo come l'ultima colonna ci riporta la “cronologia” dei nostri risparmi, tenendo la “traccia della loro storia”.

Il programma funziona anche se introduciamo importi negativi, cioè se oltre che ad aggiungere nuovi risparmi parte di quelli presenti vengono prelevati e... spesi!

- 1 L'istruzione di iterazione (o ciclo)
 - Esempi di istruzione di iterazione
 - Codifica dell'iterazione
- 2 L'iterazione indefinita
 - Esempio #1: Mi compro la moto!
 - Il tracciamento delle variabili
- 3 L'iterazione definita
 - Esempio #2: Potenza n-esima di un numero
 - Esercizio #1: Pseudocodifica di Esempio #2 (1)
 - Esercizio #2: Pseudocodifica di Esempio #2 (2)

L'iterazione definita è molto più articolata da codificare in quanto sono previste diverse possibilità che rendono questa istruzione particolarmente comoda: ricordiamo che può essere utilizzata solo quando conosciamo a priori il numero di volte che il **ciclo** deve essere eseguito.

L'iterazione definita

La sintassi da inserire nel blocco che esegue l'**iterazione a conteggio** è composta da tre parti e ciascuna di esse è una istruzione che ha come “protagonista” una **variabile**, chiamata **variabile di conteggio**:

- **Inizializzazione** della variabile che effettua il conteggio: $x = 1$ (**valore iniziale** del **contatore**);
- **Condizione** che fa ripetere il ciclo: $x \leq 5$ (**valore finale** del **contatore**);
- **Aggiornamento** della variabile di conteggio che viene fatto a ogni ripetizione: $x = x + 1$ (**incremento** o **passo**).

Queste tre operazioni vengono eseguite ogni volta che il **ciclo** viene ripetuto.

L'iterazione definita

La sintassi da inserire nel blocco che esegue l'**iterazione a conteggio** è composta da tre parti e ciascuna di esse è una istruzione che ha come “protagonista” una **variabile**, chiamata **variabile di conteggio**:

- **Inizializzazione** della variabile che effettua il conteggio: $x = 1$ (**valore iniziale** del **contatore**);
- **Condizione** che fa ripetere il ciclo: $x \leq 5$ (**valore finale** del **contatore**);
- **Aggiornamento** della variabile di conteggio che viene fatto a ogni ripetizione: $x = x + 1$ (**incremento** o **passo**).

Queste tre operazioni vengono eseguite ogni volta che il **ciclo** viene ripetuto.

L'iterazione definita

La sintassi da inserire nel blocco che esegue l'**iterazione a conteggio** è composta da tre parti e ciascuna di esse è una istruzione che ha come “protagonista” una **variabile**, chiamata **variabile di conteggio**:

- **Inizializzazione** della variabile che effettua il conteggio: $x = 1$ (**valore iniziale** del **contatore**);
- **Condizione** che fa ripetere il ciclo: $x \leq 5$ (**valore finale** del **contatore**);
- **Aggiornamento** della variabile di conteggio che viene fatto a ogni ripetizione: $x = x + 1$ (**incremento** o **passo**).

Queste tre operazioni vengono eseguite ogni volta che il **ciclo** viene ripetuto.

L'iterazione definita

La sintassi da inserire nel blocco che esegue l'**iterazione a conteggio** è composta da tre parti e ciascuna di esse è una istruzione che ha come “protagonista” una **variabile**, chiamata **variabile di conteggio**:

- **Inizializzazione** della variabile che effettua il conteggio: $x = 1$ (**valore iniziale** del **contatore**);
- **Condizione** che fa ripetere il ciclo: $x \leq 5$ (**valore finale** del **contatore**);
- **Aggiornamento** della variabile di conteggio che viene fatto a ogni ripetizione: $x = x + 1$ (**incremento** o **passo**).

Queste tre operazioni vengono eseguite ogni volta che il **ciclo** viene ripetuto.

L'iterazione definita

La sintassi da inserire nel blocco che esegue l'**iterazione a conteggio** è composta da tre parti e ciascuna di esse è una istruzione che ha come “protagonista” una **variabile**, chiamata **variabile di conteggio**:

- **Inizializzazione** della variabile che effettua il conteggio: $x = 1$ (**valore iniziale** del **contatore**);
- **Condizione** che fa ripetere il ciclo: $x \leq 5$ (**valore finale** del **contatore**);
- **Aggiornamento** della variabile di conteggio che viene fatto a ogni ripetizione: $x = x + 1$ (**incremento** o **passo**).

Queste tre operazioni vengono eseguite ogni volta che il **ciclo** viene ripetuto.

L'iterazione definita

Esempio #2: Potenza n -esima di un numero

Scriviamo un programma che legge un numero $n \leq 16$ e calcola la potenza n -esima del numero 2 utilizzando la sola operazione di moltiplicazione.

Suggerimento: per calcolare la potenza di un numero utilizzando la sola moltiplicazione è necessario moltiplicare il numero per se stesso per quante volte viene richiesto dal numero n inserito.

Ad esempio, $2^5 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 32$, quindi faremo per 5 volte il prodotto di un numero per 2.

L'iterazione definita

Esempio #2: Potenza n-esima di un numero

Dopo aver letto il numero in ingresso, utilizzeremo proprio il numero letto come valore finale della condizione del ciclo che al suo interno esegue la moltiplicazione per 2.

Con questo esempio possiamo osservare che è possibile utilizzare una **variabile** come **condizione di conteggio**.

L'iterazione definita

Esempio #2: Potenza n-esima di un numero

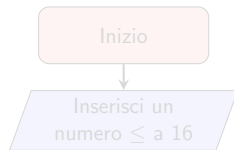
Dopo aver letto il numero in ingresso, utilizzeremo proprio il numero letto come valore finale della condizione del ciclo che al suo interno esegue la moltiplicazione per 2.

Con questo esempio possiamo osservare che è possibile utilizzare una **variabile** come **condizione di conteggio**.

L'iterazione definita

Esempio #2: Potenza n-esima di un numero

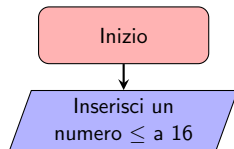
Inizializziamo una **variabile** potenza = 1 che conterrà il risultato desiderato: tale **variabile** viene inserita in un ciclo che la moltiplica per 2 tante volte quante indicate dal numero letto in **INPUT**: $\text{potenza} = \text{potenza} * 2$



L'iterazione definita

Esempio #2: Potenza n-esima di un numero

Inizializziamo una **variabile** potenza = 1 che conterrà il risultato desiderato: tale **variabile** viene inserita in un ciclo che la moltiplica per 2 tante volte quante indicate dal numero letto in **INPUT**:
 $\text{potenza} = \text{potenza} * 2$

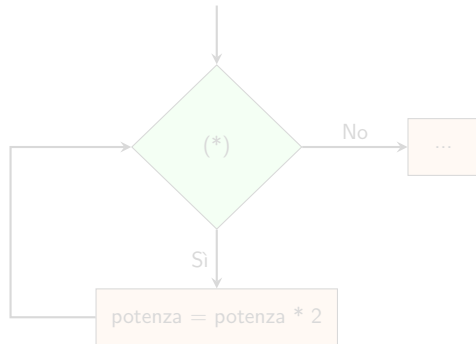


L'iterazione definita

Esempio #2: Potenza n-esima di un numero

Inizializziamo una **variabile** `potenza = 1` che conterrà il risultato desiderato: tale **variabile** viene inserita in un ciclo che la moltiplica per 2 tante volte quante indicate dal numero letto in **INPUT**: `potenza = potenza * 2`

(*) `x = 1; x <= numero; x = x + 1`

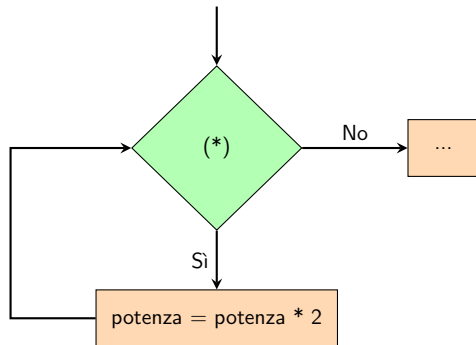


L'iterazione definita

Esempio #2: Potenza n-esima di un numero

Inizializziamo una **variabile** `potenza = 1` che conterrà il risultato desiderato: tale **variabile** viene inserita in un ciclo che la moltiplica per 2 tante volte quante indicate dal numero letto in **INPUT**: `potenza = potenza * 2`

(*) `x = 1; x <= numero; x = x + 1`



L'iterazione definita

Esercizio #1: Pseudocodifica di Esempio #2 (1)

Si consideri l'Esempio #2: Potenza n -esima di un numero.

Si scriva la corrispondente codifica in pseudocodifica formale.

L'iterazione definita

Esercizio #2: Pseudocodifica di Esempio #2 (2)

Si consideri l'Esempio #2: Potenza n -esima di un numero.

Per completare il programma è necessario introdurre un controllo sul dato inserito dall'utente affinché non superi il valore di 16, come indicato dal testo.

Si modifichino il diagramma di flusso fornito dal docente e la pseudocodifica formale scritta nell'Esercizio #1 aggiungendo il controllo richiesto.

- [1] P. Camagni e R. Nikolassy, *TeknoLab: Tecnologie informatiche E Laboratorio: Office automation, Sviluppo del Pensiero computazionale, scratch e applicazioni per arduino, app inventor per android, Algobuild e Codifica C++*. Hoepli, 2017.