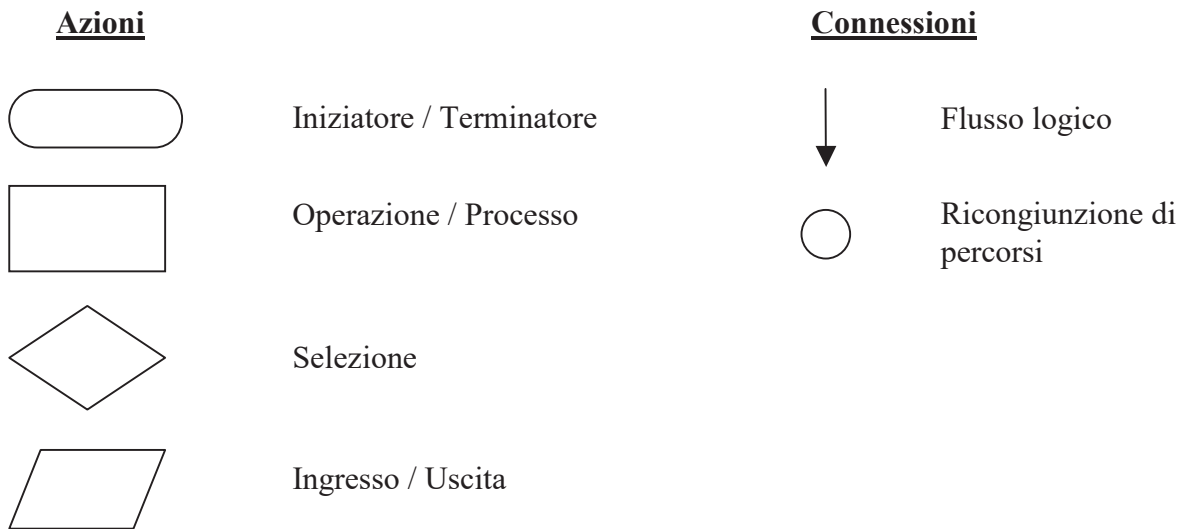


Rappresentazione con i diagrammi di flusso (Flow - chart)

Questo tipo di rappresentazione grafica degli algoritmi, sviluppato negli anni '50, utilizza una serie di simboli grafici dal contenuto evocativo che richiamano visivamente le principali operazioni all'interno di un algoritmo.

Un flow - chart si costruisce combinando in modo opportuno i simboli grafici disponibili, in modo tale da riprodurre fedelmente l'evoluzione dell'algoritmo in tutte le sue diramazioni e possibilità.

Introduciamo i principali simboli che verranno utilizzati per la rappresentazione degli algoritmi mediante i diagrammi di flusso.



Caratteristiche della rappresentazione con i diagrammi di flusso

Vantaggi

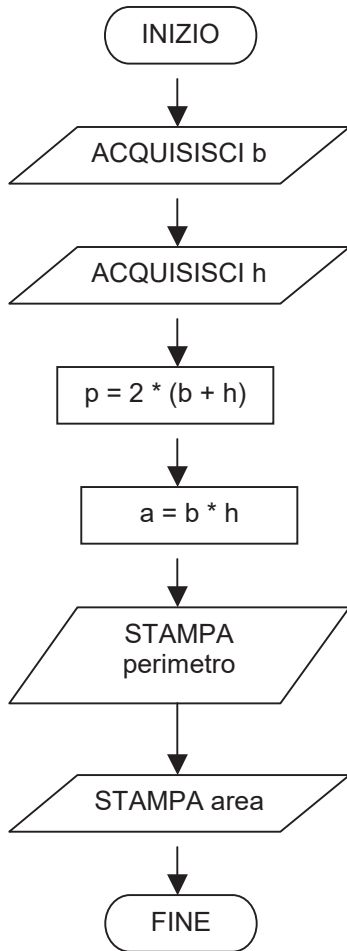
- Permette di realizzare diagrammi intuitivi e di facile lettura.

Svantaggi

- Incapacità di sintetizzare algoritmi complessi;
- Rappresentazione non immediatamente trasferibile in un linguaggio di programmazione.

Esempio 1

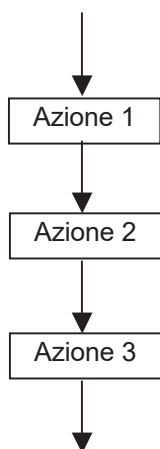
Descrivere utilizzando i diagrammi di flusso l'algoritmo che dati in ingresso il valore della base e dell'altezza di un rettangolo ne calcoli il perimetro e l'area.



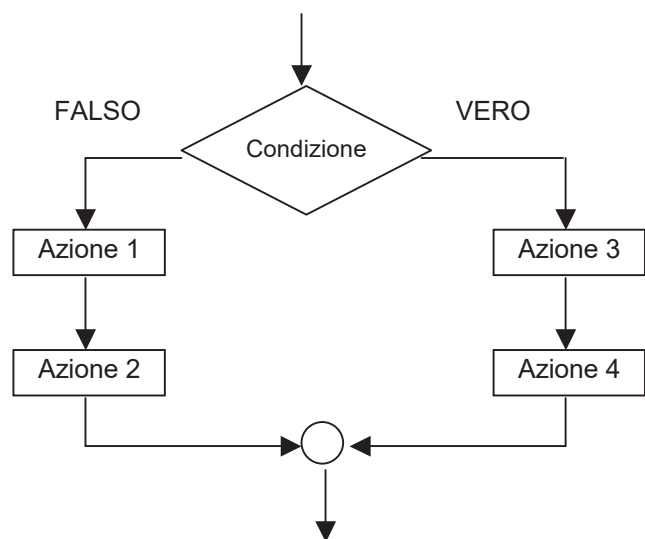
Descrizione in linguaggio naturale

1. Inizio.
2. Acquisisci il valore della base b.
3. Acquisisci il valore dell'altezza h.
4. Calcola il perimetro del rettangolo $p = 2 * (b + h)$.
5. Calcola l'area del rettangolo $a = b * h$.
6. Stampa il valore del perimetro.
7. Stampa il valore dell'area.
8. Fine.

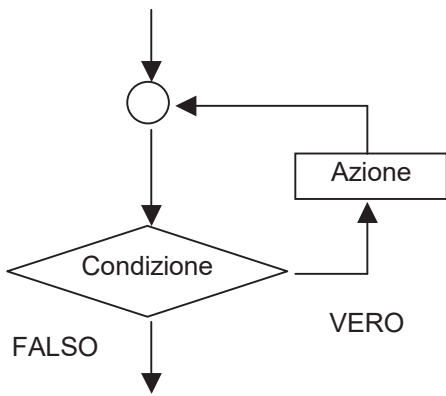
Strutture base dei diagrammi di flusso



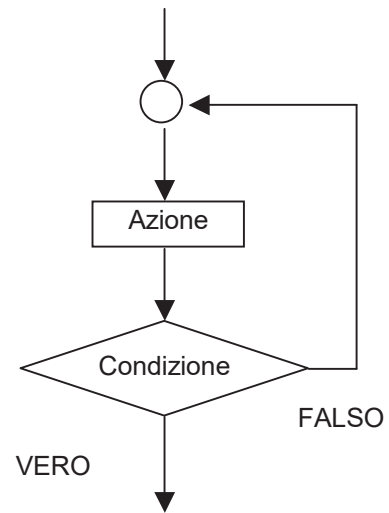
Sequenza



Blocco di selezione



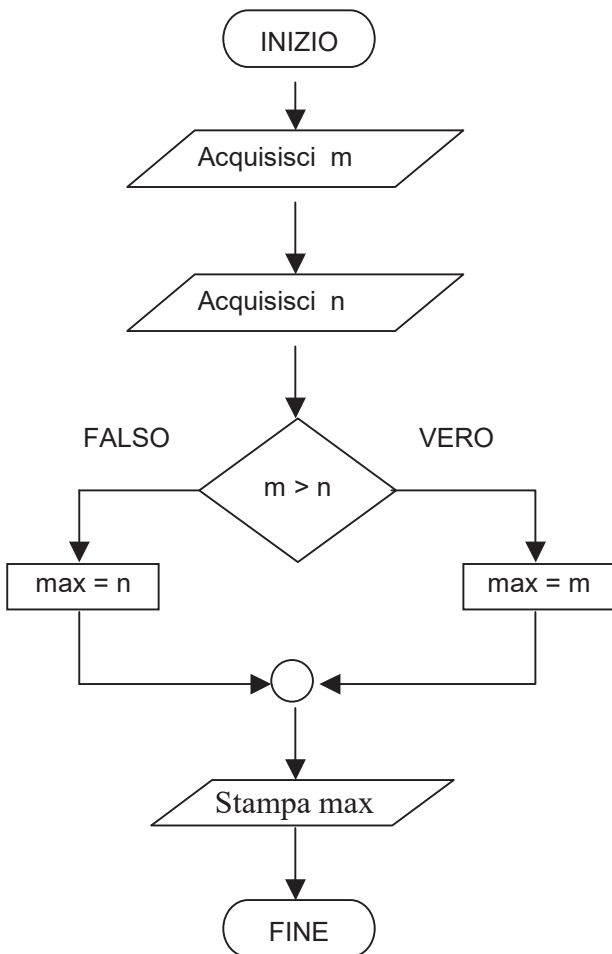
Ciclo While – Do



Ciclo Repeat - Until

Esempio 2

Descrivere utilizzando i diagrammi di flusso l’algoritmo che determina il valore maggiore tra due numeri m e n.



Descrizione in linguaggio naturale

1. Inizio.
2. Acquisisci m.
3. Acquisisci n.
4. Se $m > n$ allora $max = m$ altrimenti $max = n$.
5. Stampa max.
6. Fine.

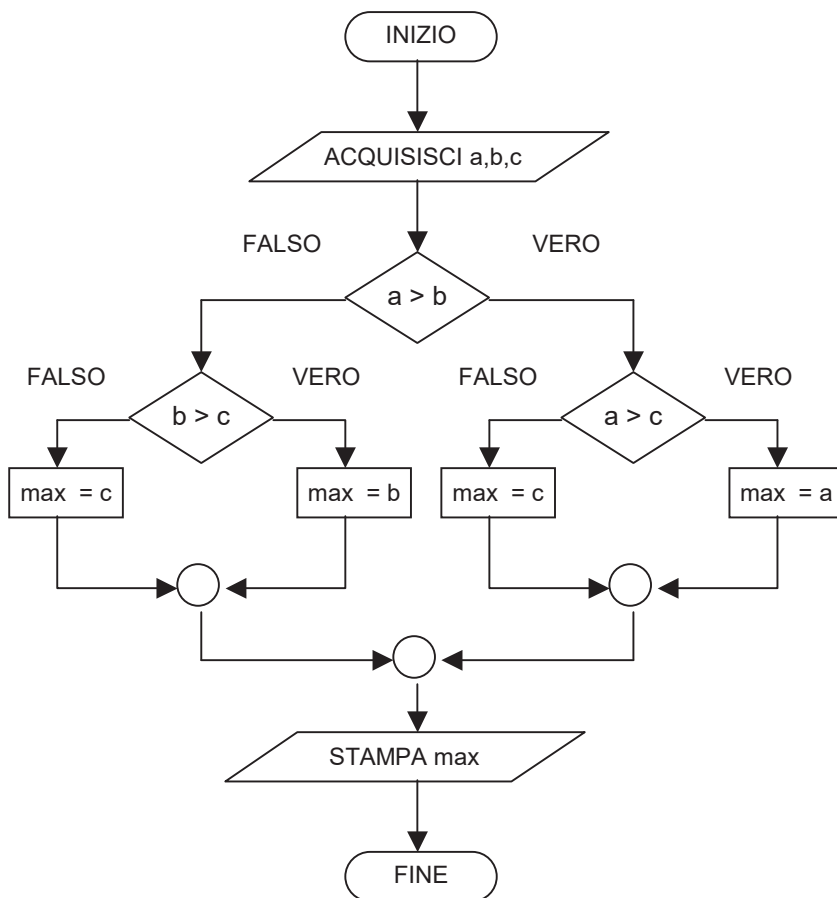
Il modo con cui questo esecutore risolve il problema è molto diverso da quello seguito dall'esecutore umano. Questo esempio mette in luce che la struttura e la sequenza di operazioni che descrivono l'algoritmo sono funzione dell'esecutore.

In particolare vengono messe in evidenza:

- le operazioni elementari in grado di eseguire;
- comprensione dell'algoritmo.

Esempio 3

Descrivere utilizzando i diagrammi di flusso l'algoritmo che determina il valore maggiore tra tre numeri a, b, c.



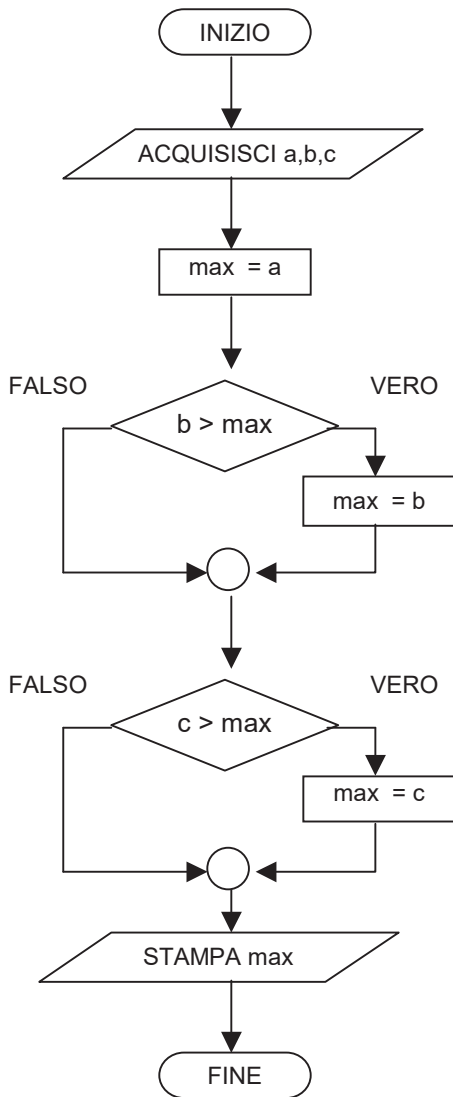
Descrizione in linguaggio naturale

1. Inizio.
2. Acquisisci i valori di a,b,c.
3. Se $a > b$ allora prosegui al passo 6.
4. Se $b > c$ allora $\max = a$ altrimenti $\max = c$.
5. Salta al passo 7.
6. Se $a > c$ allora $\max = b$ altrimenti $\max = c$.
7. Stampa il valore massimo.
8. Fine.

Osservazioni

1. Non è possibile mettere in collegamento i rami di due blocchi di selezione sebbene richiedono l'esecuzione della stessa operazione.
2. Per verificare che l'algoritmo funziona correttamente si devono fare delle simulazioni del suo funzionamento utilizzando dei valori di prova di cui si conoscano le soluzioni.
3. Se esistono due rami dei blocchi di selezione con la stessa azione, probabilmente l'algoritmo che abbiamo realizzato non è il più efficiente (migliore).

Soluzione migliore

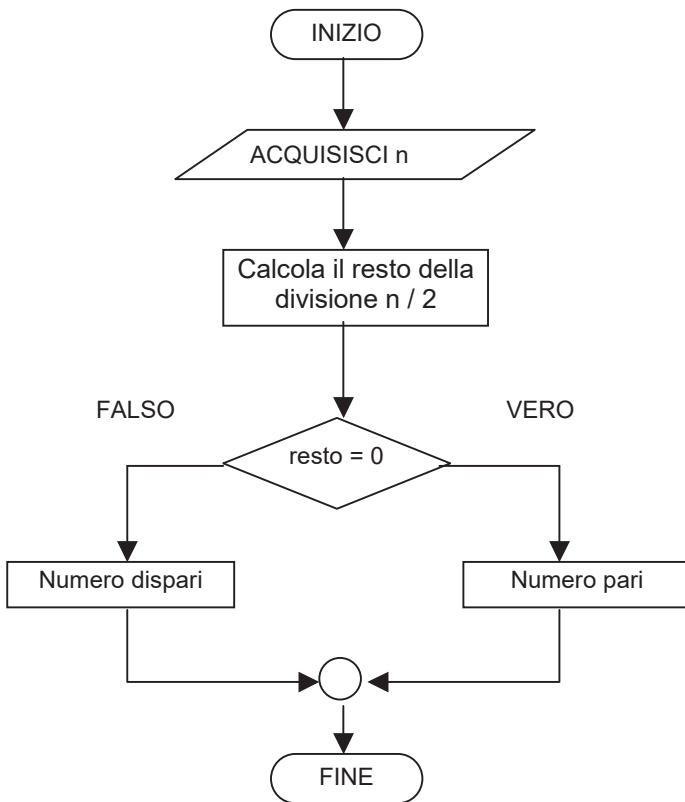


Descrizione in linguaggio naturale

1. Inizio.
2. Acquisisci i valori a,b,c.
3. Poni max = a.
4. Se $b > \text{max}$ allora poni max = b.
5. Se $c > \text{max}$ allora poni max = c.
6. Stampa il valore massimo.
7. Fine.

Esempio 4

Descrivere utilizzando i diagrammi di flusso l'algoritmo che determina se un numero intero è pari o dispari.



Descrizione in linguaggio naturale

1. Inizio.
2. Acquisisci il numero intero n.
3. Calcola il resto della divisione $n / 2$.
4. Se il resto = 0 allora il numero è pari altrimenti è dispari.
5. Fine.

Caratteristiche di un algoritmo

Affinché un insieme di operazioni costituisca un algoritmo deve soddisfare i requisiti seguenti:

- il numero di operazioni svolte nella soluzione deve essere finito e inoltre ogni singola operazione deve essere eseguita in un intervallo di tempo determinato;
- ogni operazione deve essere descritta in modo preciso e non ambiguo;
- deve essere deterministico ovvero, a partire dagli stessi dati iniziali, l'insieme delle operazioni deve sempre produrre i medesimi risultati;
- deve essere eseguibile, ovvero deve esistere un esecutore, automatico o manuale, in grado di effettuare le operazioni che definiscono l'algoritmo;
- deve essere comprensibile al suo esecutore
- deve essere generale e quindi in grado di risolvere non un singolo problema, ma tutti quelli di una determinata classe.