

Procedimento di risoluzione di un algoritmo

Sequenza

Progettare un algoritmo che permetta di effettuare la somma di tre numeri interi.

Tabella dell'algoritmo¹

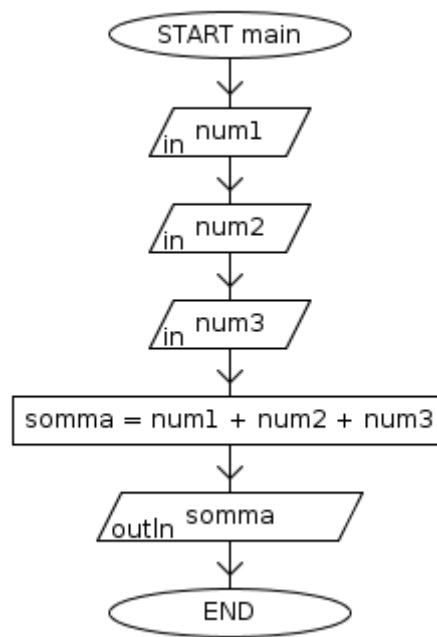
Domanda	Risposta
Chi lo usa?	Un utente
Obiettivo	Ottenere la somma di tre numeri
Input	Tre numeri
Output	La somma dei tre numeri
Procedimento	1. Chiedere i tre numeri all'utente 2. Sommare tre numeri 3. Stampare il numero che rappresenta la somma
Casi limite	Nessuno, eventualmente overflow
Test programmato	$5 + 5 + 5 = 15$ $10 + 12 + (-4) = 18$ $7 + (-3) + 0 = 4$ $-8 + 0 + 5 = -3$

Tabella delle variabili

Nome	Tipo	Descrizione	I/O/L	Vincoli
num1	int	Primo addendo	I	-
num2	int	Secondo addendo	I	-
num3	int	Terzo addendo	I	-
somma	int	Somma dei tre numeri	O	-

¹ [Fabio Biscaro](#)

Diagramma di flusso (flow-chart)



Test

num1	num2	num3	somma
I: 5	-	-	-
5	I: 5	-	-
5	5	I: 5	-
5	5	5	15
5	5	5	O: 15

num1	num2	num3	somma
I: 10	-	-	-
10	I: 12	-	-
10	12	I: -4	-
10	12	-4	18
10	12	-4	O: 18

num1	num2	num3	somma
I: 7	-	-	-
7	I: -3	-	-
7	-3	I: 0	-
7	-3	0	4
7	-3	0	O: 4

num1	num2	num3	somma
I: -8	-	-	-
-8	I: 0	-	-
-8	0	I: 5	-
-8	0	5	-3
-8	0	5	O: -3

Progettare un algoritmo che calcoli l'area di un triangolo equilatero.

Tabella dell'algoritmo²

Domanda	Risposta
Chi lo usa?	Un utente
Obiettivo	Visualizzare l'area di un triangolo equilatero
Input	base e altezza del triangolo
Output	area del triangolo
Procedimento	1. Chiedere la base 2. Chiedere l'altezza 3. Calcolare l'area ($\text{area} = (\text{base} * \text{altezza}) / 2$) 4. Visualizzare l'area
Casi limite	• $\text{base} \leq 0$ • $\text{altezza} \leq 0$ • $\text{altezza} \neq \sqrt{\text{base}^2 + (\text{base}/2)^2}$
Test programmato	base = 10, altezza = 10, area = 50 (non è un triangolo equilatero) base = 10, altezza = 11.18, area = 55.9 base = -10, altezza = 11.18 area = -55.9 (sbagliata la base) base = 10, altezza = 0 area = -55.9 (sbagliata l'altezza)

Tabella delle variabili

Nome	Tipo	Descrizione	I/O/L	Vincoli

² [Fabio Biscaro](#)

Diagramma di flusso (flow-chart)

...

Test

base	altezza	area

Condizione

Progettare un algoritmo che prenda in input un numero intero e visualizzi la scritta **pari** se il numero è pari, e **dispari** se il numero è dispari.

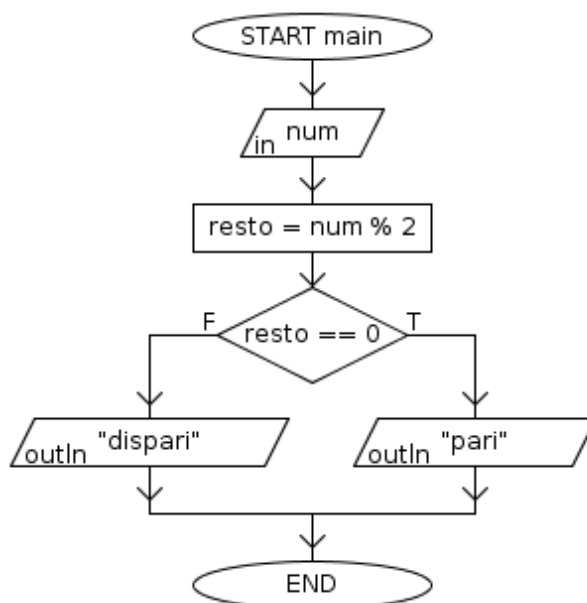
Tabella dell'algoritmo

Domanda	Risposta
Chi lo usa?	Un utente
Obiettivo	Determinare se un numero è pari o dispari
Input	Il numero
Output	«pari» o «dispari»
Procedimento	4. Chiedere all'utente un numero 5. Calcolo il resto della divisione per due 6. Se il resto è zero scrivo «pari», altrimenti scrivo «dispari»
Casi limite	Nessuno
Test programmato	12 è pari, 11 è dispari

Tabella delle variabili

Nome	Tipo	Descrizione	I/O	Vincoli
num	int	Numero da valutare	I	-
resto	int	Resto della divisione del numero per 2	L	-

Diagramma di flusso (flow-chart)



Progettare un algoritmo che, dati due numeri interi x e y, calcoli il risultato e il resto della divisione intera tra x e y.

Tabella dell'algoritmo³

Domanda	Risposta
Chi lo usa?	Un utente
Obiettivo	Ottenere il quoziente dei due numeri interi e il resto
Input	Due numeri interi
Output	Quoziente e resto
Procedimento (se è chiaro il funzionamento del flow-chart, il procedimento può essere omissis)	1. Chiedere i due numeri all'utente 2. Fare la divisione intera dei due numeri ottenendo il quoziente 3. Moltiplico il divisore per il quoziente (divisore approssimato) 4. Sottrarre al dividendo il "divisore approssimato" (resto) 5. Visualizzare il quoziente ed il resto
Casi limite	divisore = 0
Test programmato	$9 / 3 = 3, R: 0$ $8 / 3 = 2, R: 2$

Tabella delle variabili

Nome	Tipo	Descrizione	I/O	Vincoli
dividendo	int	Il dividendo	I	-
divisore	int	Il divisore	I	divisore != 0
quoziente	int	Quoziente della divisione	O	-
resto	int	Resto della divisione	O	-

³ [Fabio Biscaro](#)

Diagramma di flusso (flow-chart)

