

Programmazione e Laboratorio di Programmazione

Lezione III

I diagrammi di flusso

Nozione intuitiva di algoritmo

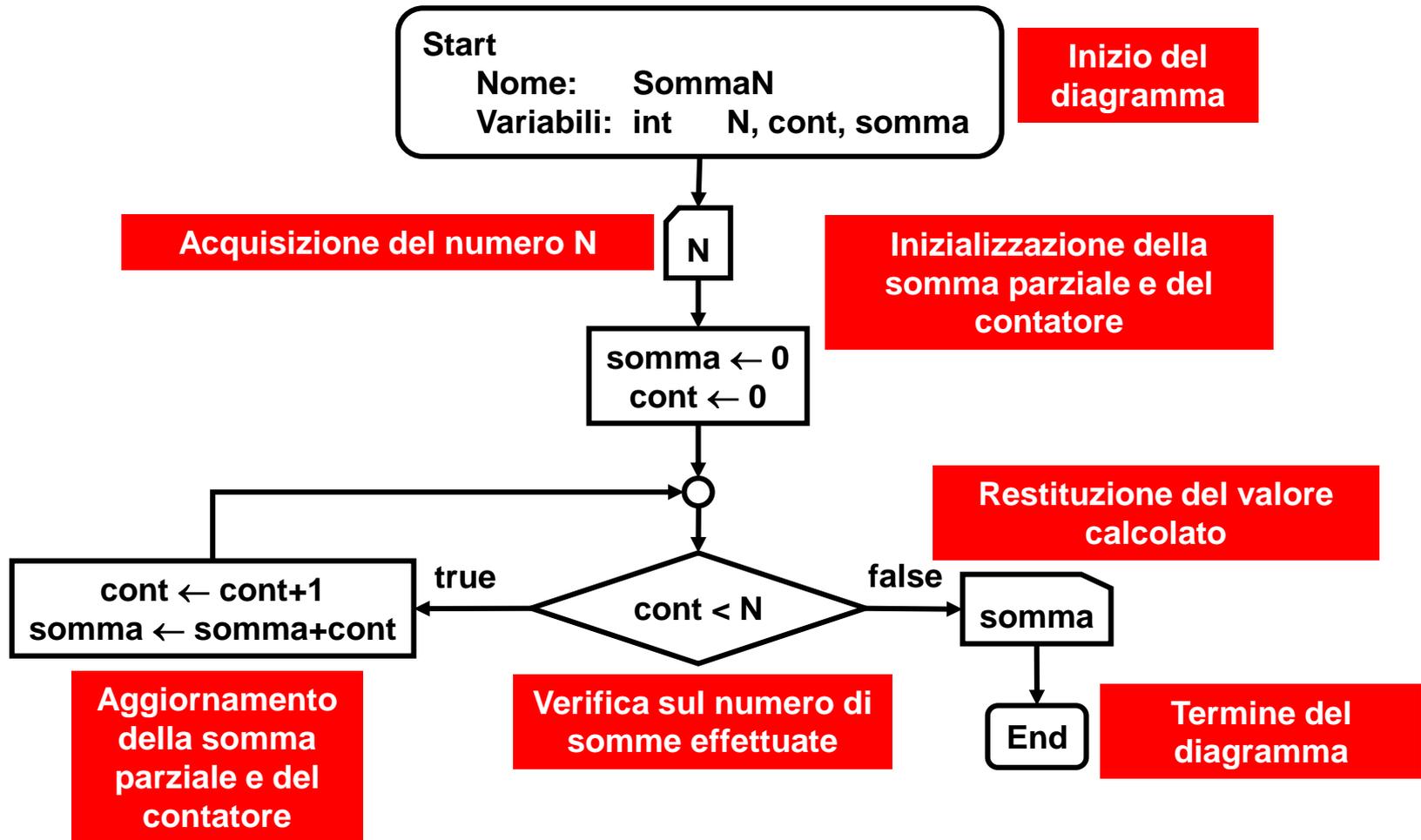
- **Nozione intuitiva di algoritmo**
 - è una sequenza finita di istruzioni
 - ogni istruzione è una stringa di lunghezza finita costruita a partire da un alfabeto di dimensione finita
 - deve esistere un agente di calcolo **C** capace di eseguire le istruzioni dell'algoritmo
 - **C** deve avere capacità di memorizzazione
 -

I diagrammi di flusso

- **Diagrammi di Flusso:**
 - un formalismo grafico per la descrizione di algoritmi
 - un particolare simbolo grafico detto blocco è associato ad ogni tipo di operazione
 - i blocchi sono collegati tra loro da archi che definiscono l'ordine di esecuzione delle istruzioni

Esempio

- Calcolare la somma dei numeri interi $1 \leq i \leq N$



Capacità di memorizzazione

- **Modello:**

descrizione della realtà limitatamente agli aspetti di interesse

- **Modello di memoria:**

- insieme di **locazioni**
- ogni locazione può **memorizzare** un **valore** di tipo **intero**, **carattere**, o **booleano**
- una locazione o è correntemente in **uso** o è **disponibile**
 in uso
 disponibile
- locazioni correntemente in uso sono dette **variabili**
- ogni variabile è identificata da un **nome** e da un **tipo** (il tipo del valore memorizzabile)

	intero	
A	3	✓
	carattere	
B	'c'	✓
	booleano	
C	true	✓

Stato della Memoria

- **Molto informalmente:**

è una “foto” istantanea della memoria

- **Molto meno informalmente:**

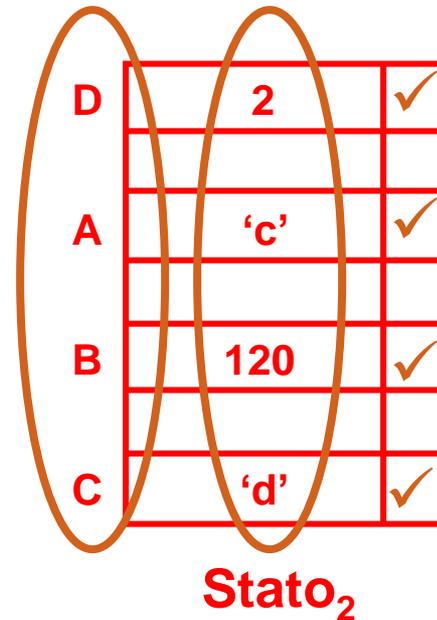
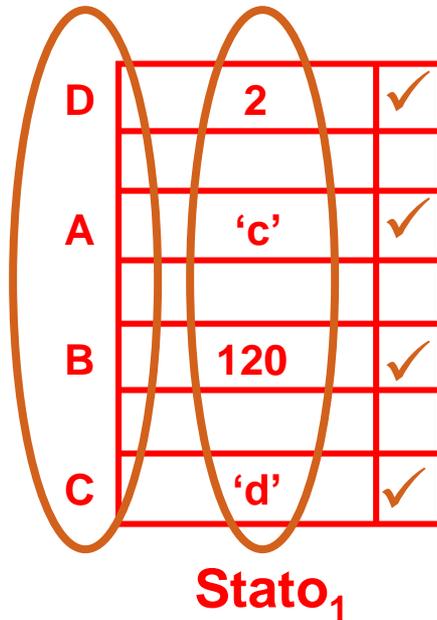
è determinato dall'insieme delle triple

(nome_{var}, tipo_{var}, valore_{var})

Stato della Memoria

- $Stato_1 = Stato_2$?

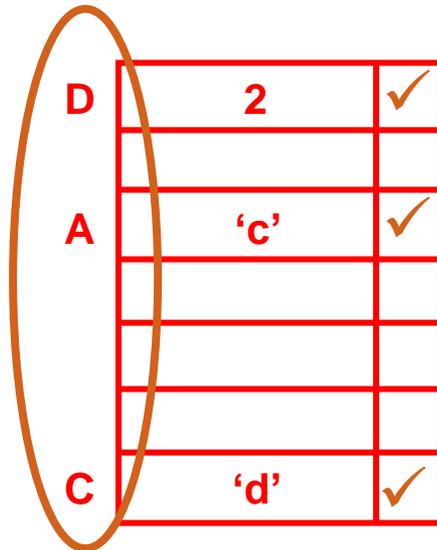
SI



Stato della Memoria

- $Stato_1 = Stato_2$?

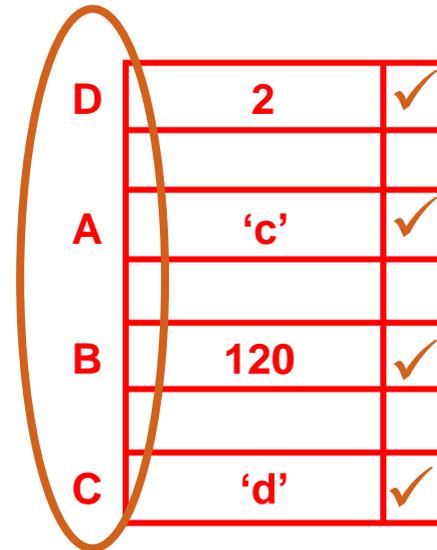
NO



A diagram representing memory state 1. It consists of a 6x3 grid of cells. The first column contains labels 'D', 'A', and 'C' at the top, bottom, and bottom of a vertical oval. The second column contains the value '2' in the first row, 'c' in the second row, and 'd' in the sixth row. The third column contains checkmarks in the first, second, and sixth rows. All other cells are empty.

D	2	✓
A	'c'	✓
C	'd'	✓

$Stato_1$



A diagram representing memory state 2. It consists of a 6x3 grid of cells. The first column contains labels 'D', 'A', 'B', and 'C' at the top, middle, middle, and bottom of a vertical oval. The second column contains the value '2' in the first row, 'c' in the second row, '120' in the fifth row, and 'd' in the sixth row. The third column contains checkmarks in the first, second, fifth, and sixth rows. All other cells are empty.

D	2	✓
A	'c'	✓
B	120	✓
C	'd'	✓

$Stato_2$

Stato della Memoria

- **Stato₁ = Stato₂ ?**

NO

D	2	✓
A	'c'	✓
B	120	✓
C	'd'	✓

Stato₁

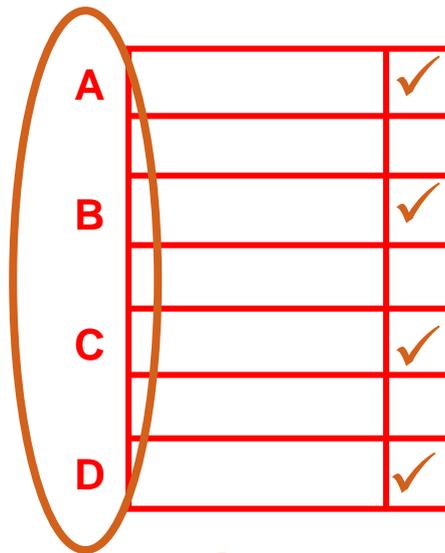
D	8	✓
A	12	✓
B	'm'	✓
C	'd'	✓

Stato₂

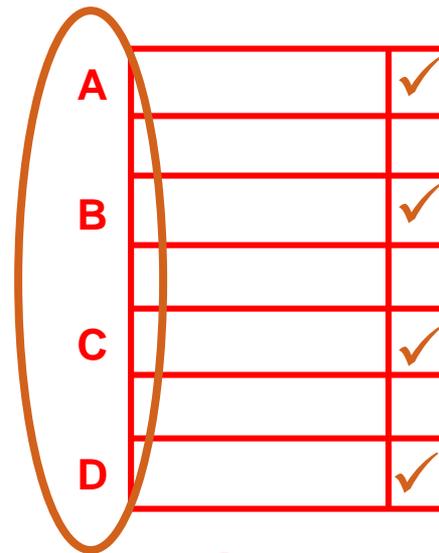
Stato della Memoria

- **Stato₁ = Stato₂ ?**

Non lo so



Stato₁



Stato₂

Stato della Memoria

- **Stato₁ = Stato₂ ?**

SI

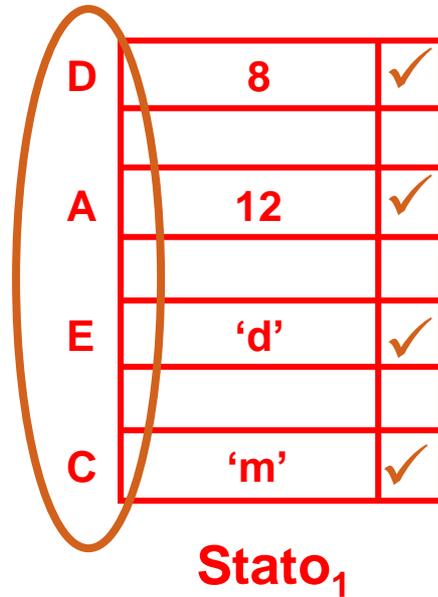


Diagram illustrating the memory state **Stato₁**. It shows a table with four rows, each representing a memory location (D, A, E, C). The values are 8, 12, 'd', and 'm' respectively. Each row has a checkmark in the third column, indicating that the value is correct. The entire table is enclosed in a red oval.

D	8	✓
A	12	✓
E	'd'	✓
C	'm'	✓

Stato₁

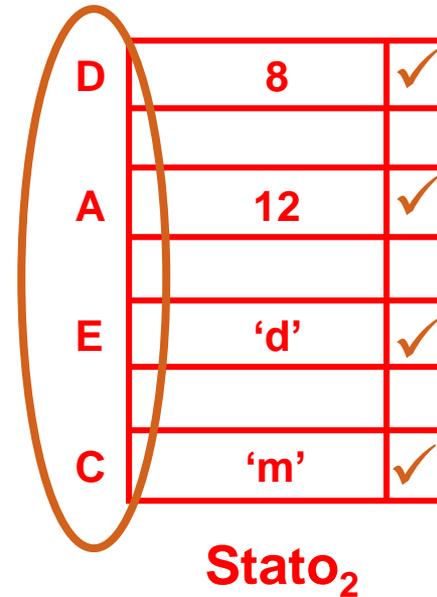


Diagram illustrating the memory state **Stato₂**. It shows a table with four rows, each representing a memory location (D, A, E, C). The values are 8, 12, 'd', and 'm' respectively. Each row has a checkmark in the third column, indicating that the value is correct. The entire table is enclosed in a red oval.

D	8	✓
A	12	✓
E	'd'	✓
C	'm'	✓

Stato₂

Stato della Memoria

- $Stato_1 = Stato_2$?

SI

D	8	✓
A	12	✓
B	true	✓
C	'd'	✓

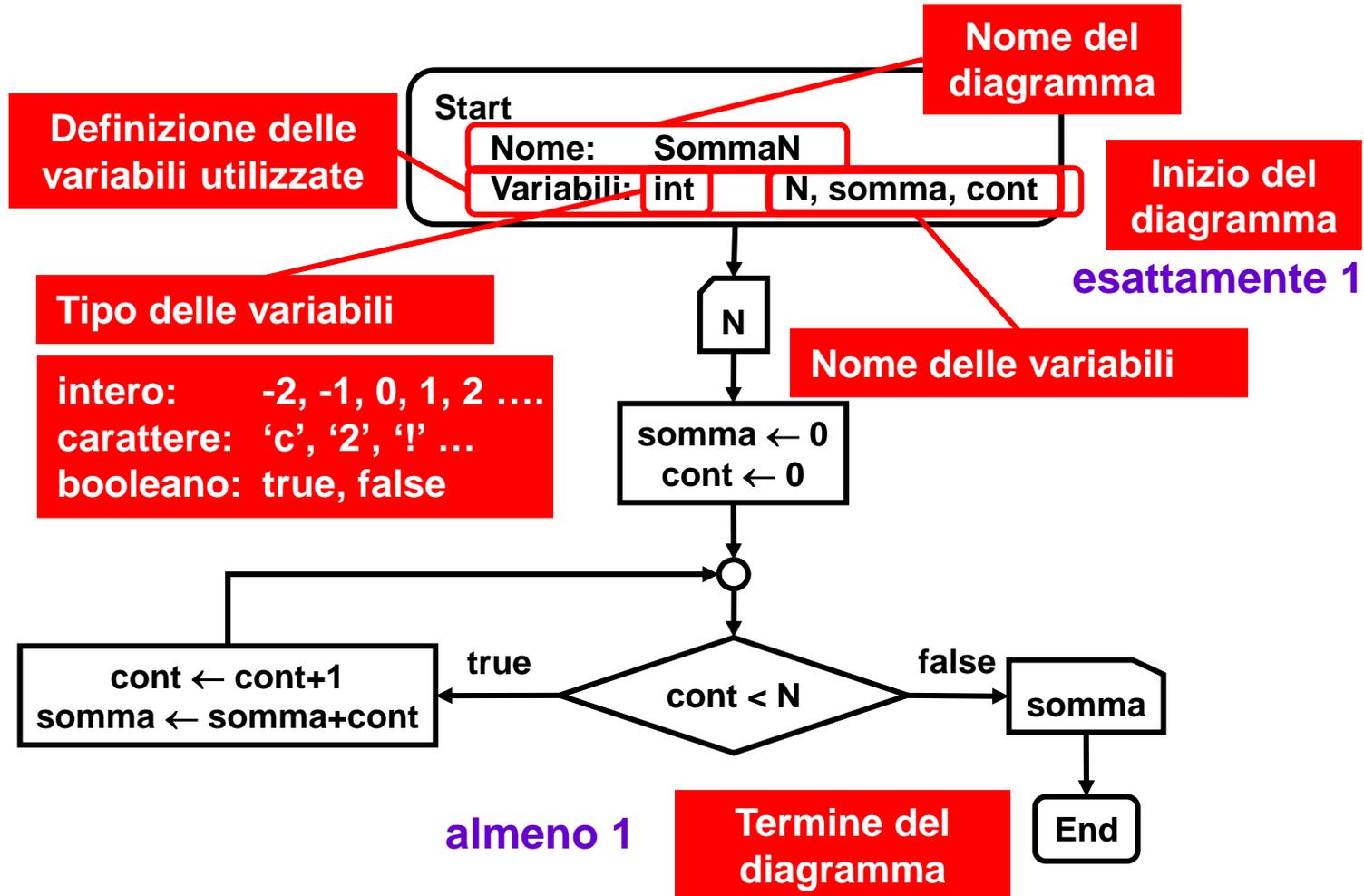
Stato₁

A	12	✓
D	8	✓
B	true	✓
C	'd'	✓

Stato₂

Tipologia dei blocchi

- **Blocchi di inizio e di termine**



Definizione di una variabile

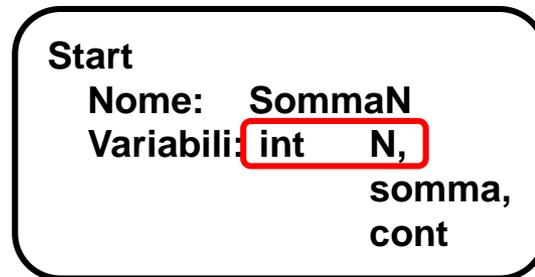
- **Blocco di inizio**

Per ognuna delle variabili elencate nel blocco

1. si individua una locazione di memoria disponibile
2. si riserva tale locazione
3. si associano a tale locazione il **nome** e il **tipo** specificati

A	2	✓
B	120	✓

Stato_I



A	2	✓
N		✓
B	120	✓

intero

Stato_F

Tipologia dei blocchi

- **Blocco di termine:**

si rilascia la memoria allocata ad ognuna delle variabili elencate nel blocco “Start”

cont	4	✓
N	4	✓
somma	10	✓

Stato_I

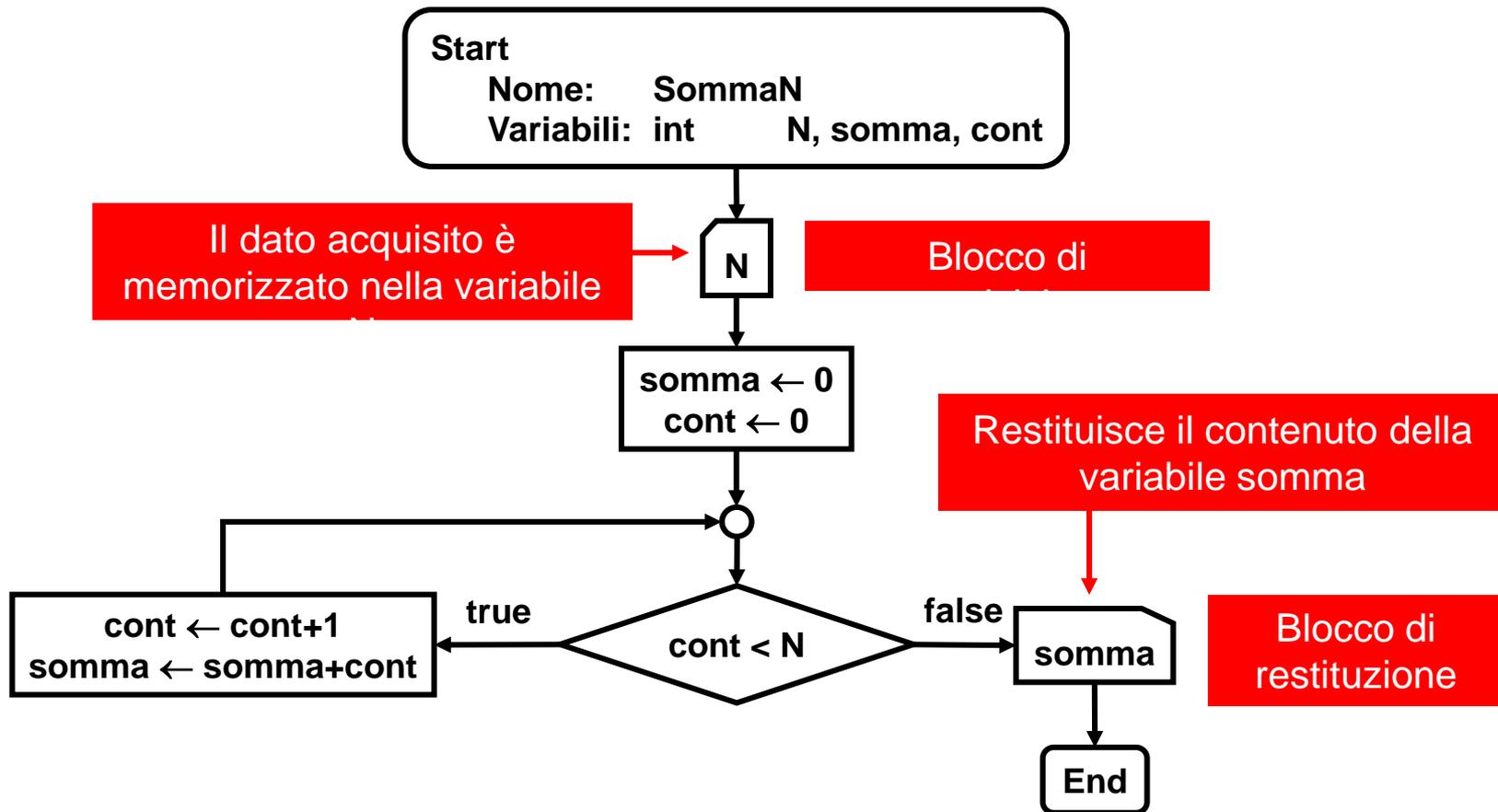
End

cont	4	
N	4	
somma	10	

Stato_F

Tipologia dei blocchi

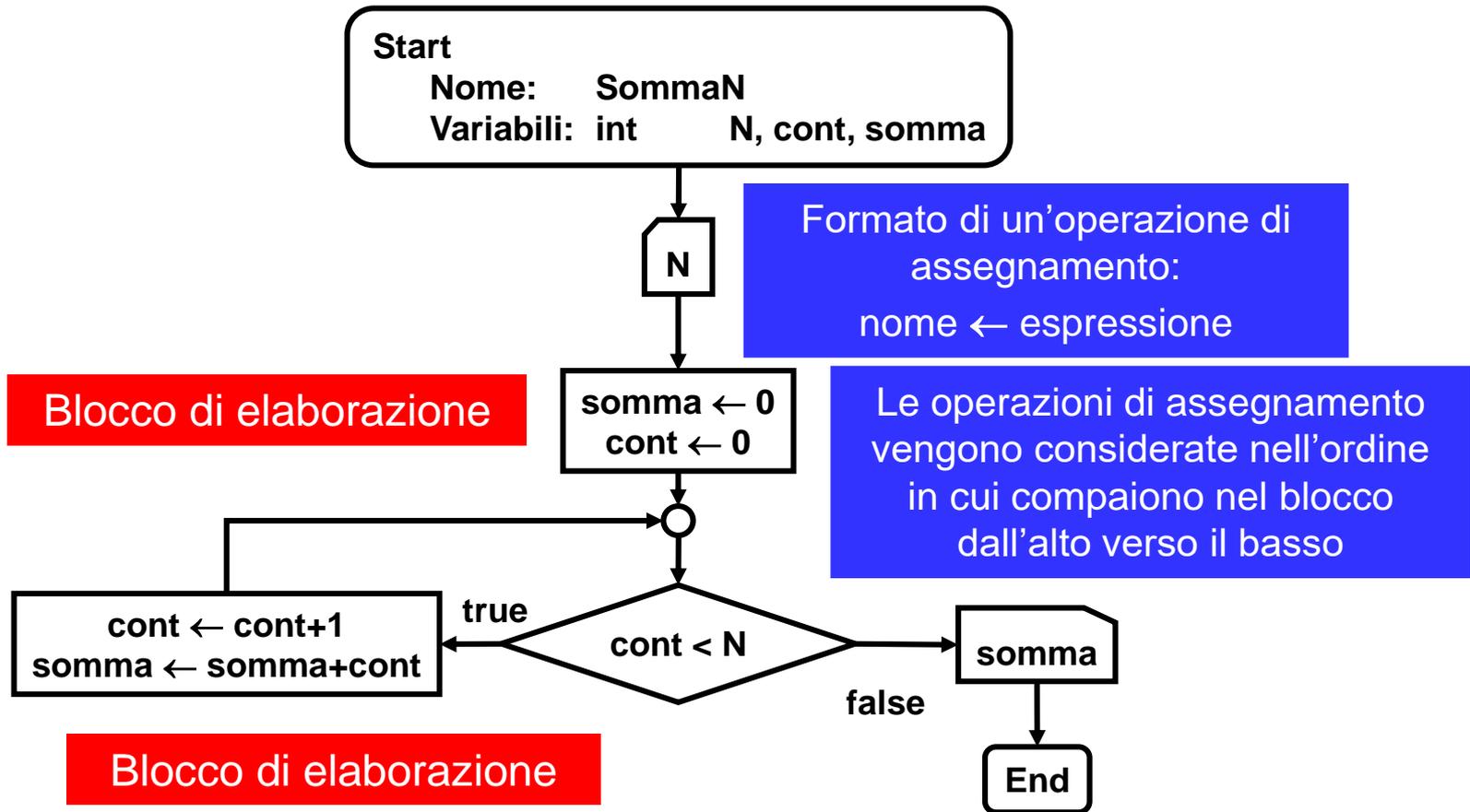
- **Blocchi di acquisizione e di restituzione dati**



Tipologia dei blocchi

- **Blocco di elaborazione**

Contiene una sequenza di operazioni di assegnamento



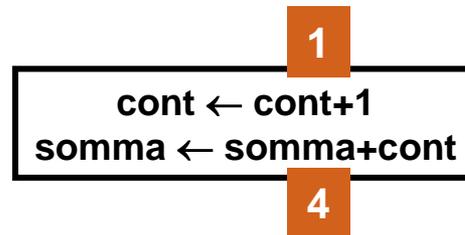
Operazioni di assegnamento

- **nome** ← **espressione**

1. Si valuta il valore di **espressione** nello stato attuale della memoria
2. Si aggiorna con tale valore il contenuto della variabile identificata da **nome**

somma	3	✓
N	3	✓
cont	0	✓

Stato_I

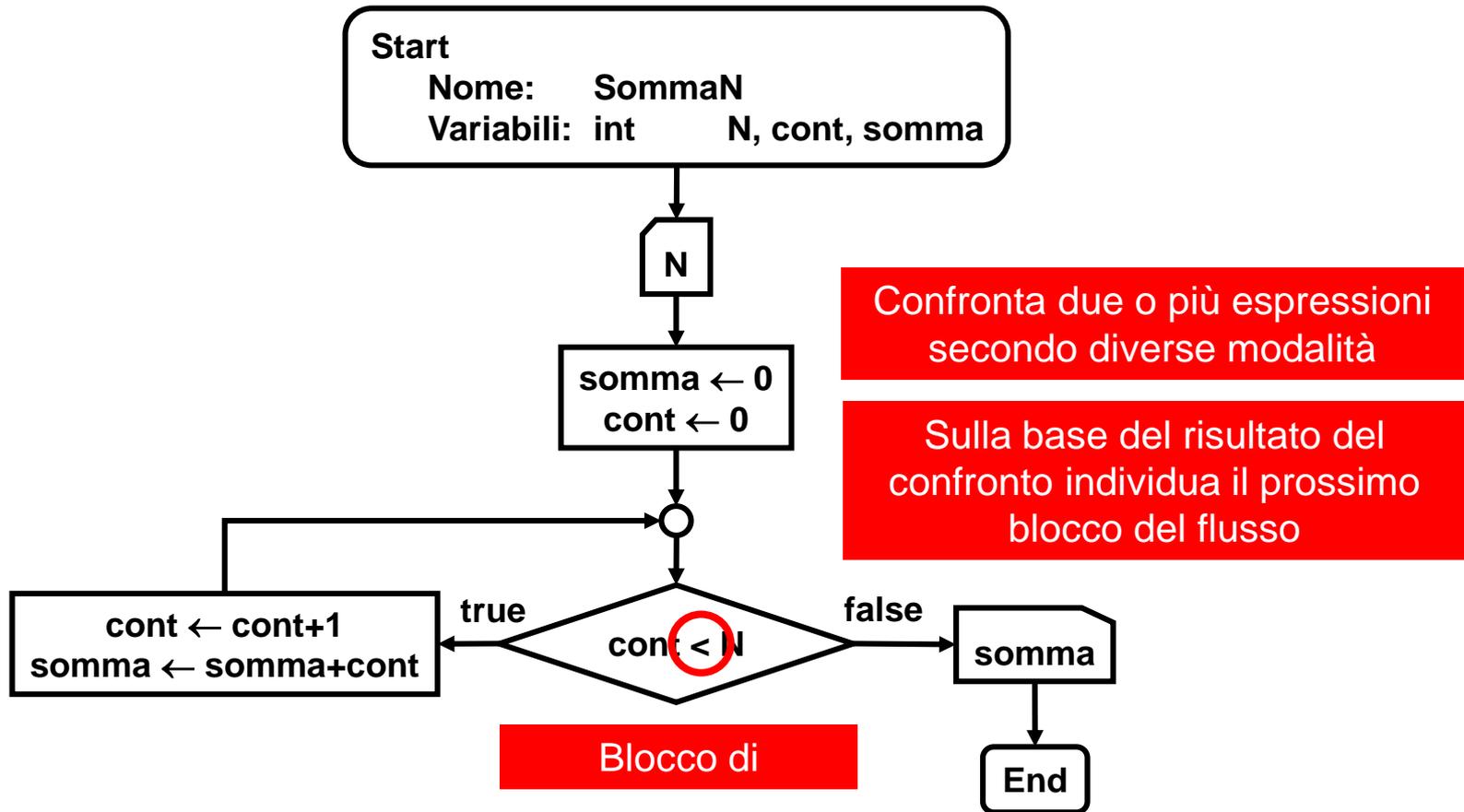


somma	4	✓
N	3	✓
cont	1	✓

Stato_F

Tipologia dei blocchi

- **Blocco di decisione**



Riassumendo

- **Blocco di inizio**

int
bool
char

Start

Nome: nome del diagramma
Variabili: tipo₁ nome₁
 tipo₂ nome₂

- **Blocco di termine**

End

- **Blocco di acquisizione**

nome₁, nome₂, ...

- **Blocco di restituzione**

nome₁, nome₂, ...

- **Blocco di elaborazione**

nome₁ ← espressione₁
nome₂ ← espressione₂
.....

Riassumendo

- **Blocco di decisione**



Condizioni di validità

- **Esiste un solo blocco di inizio e almeno un blocco di termine;**
- **il blocco di inizio ha un solo arco uscente e non ha archi entranti;**
- **ogni blocco di uscita ha un solo arco entrante e non ha archi uscenti;**
- **ciascun blocco di elaborazione, di acquisizione, e di restituzione ha un solo arco entrante e un solo arco uscente;**
- **ciascun blocco di decisione ha un solo arco entrante e due uscenti;**
- **ciascun arco entra in un blocco o si innesta su un altro arco;**
- **ciascun blocco è raggiungibile dal blocco iniziale;**
- **da qualsiasi blocco è possibile raggiungere almeno uno dei blocchi di termine.**

Operatori

- **Operatori aritmetici**

- $+$: int x int \rightarrow int somma tra interi
- $-$: int x int \rightarrow int differenza tra interi
- $*$: int x int \rightarrow int prodotto tra interi
- $/$: int x int \rightarrow int divisione intera
- $\%$: int x int \rightarrow int resto della divisione intera

- **Operatori logici**

- not : bool \rightarrow bool negazione
- and : bool x bool \rightarrow bool and logico
- or : bool x bool \rightarrow bool or logico

Operatori

- **Operatori di confronto tra interi**

- **= : int x int → bool** **test di uguaglianza**
- **> : int x int → bool** **strettamente maggiore**
- **≥ : int x int → bool** **maggiore o uguale**
- **< : int x int → bool** **strettamente minore**
- **≤ : int x int → bool** **minore uguale**

Ordinamento lessicografico

Tabella dei codici ASCII

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

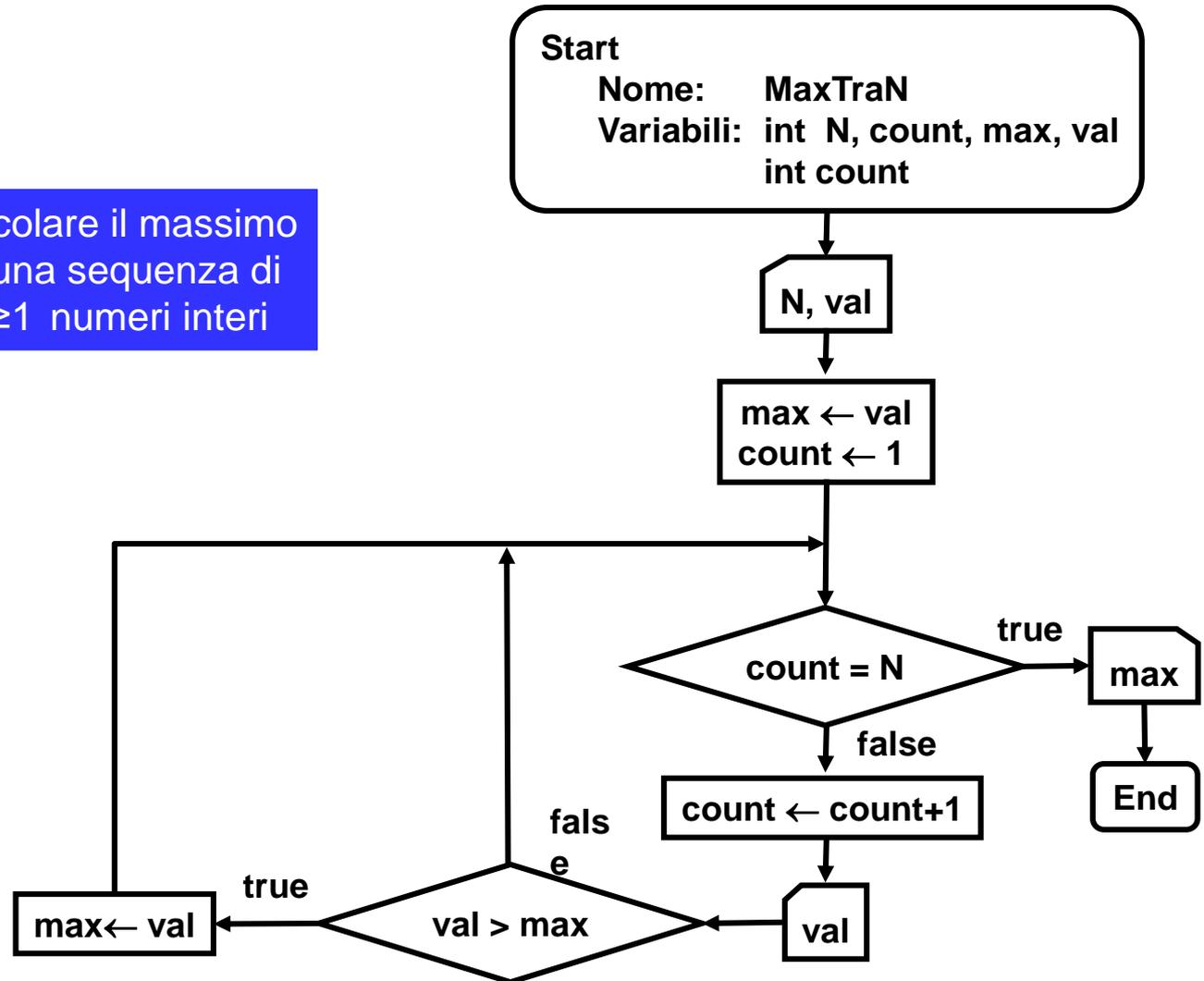
Operatori

- **Operatori di confronto tra caratteri**

- **= : char x char → bool** test di uguaglianza
- **< : char x char → bool** precede lessicograficamente
- **≤ : char x char → bool** uguale o precede
lessicograficamente
- **> : char x char → bool** segue lessicograficamente
- **≥ : char x char → bool** uguale o segue
lessicograficamente

Un semplice esempio

Calcolare il massimo
di una sequenza di
 $N \geq 1$ numeri interi



Vettori

- **Vettore (monodimensionale) di n elementi:**

definisce una corrispondenza biunivoca tra un multi-
insieme omogeneo di n elementi e l'insieme di interi
{0, 1, ..., n-1}

- **Esempio:**

Vettore di 5 interi

0	5
1	-4
2	32
3	-4
4	27

- **Definizione**

tipo_{Vettore} **nome**_{Vettore} [**dim**_{Vettore}]

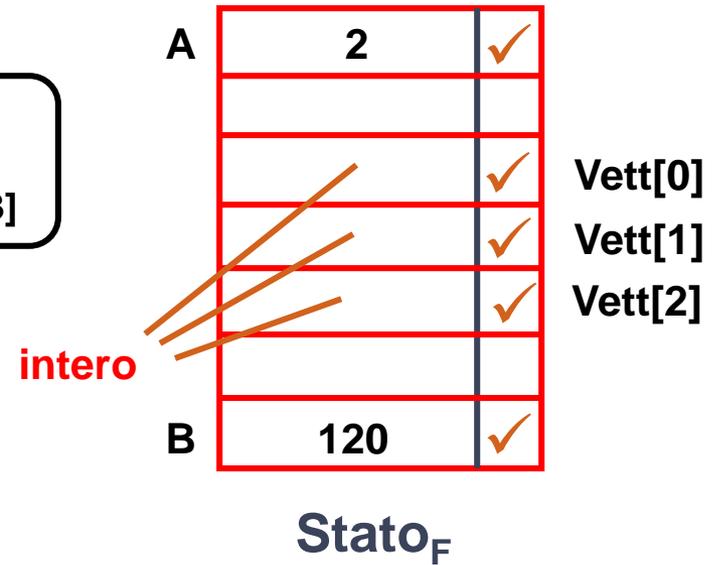
costante intera

Vettori

- Effetto



Start
...
Variabili: int Vett[3]



- Accesso all'elemento di un vettore

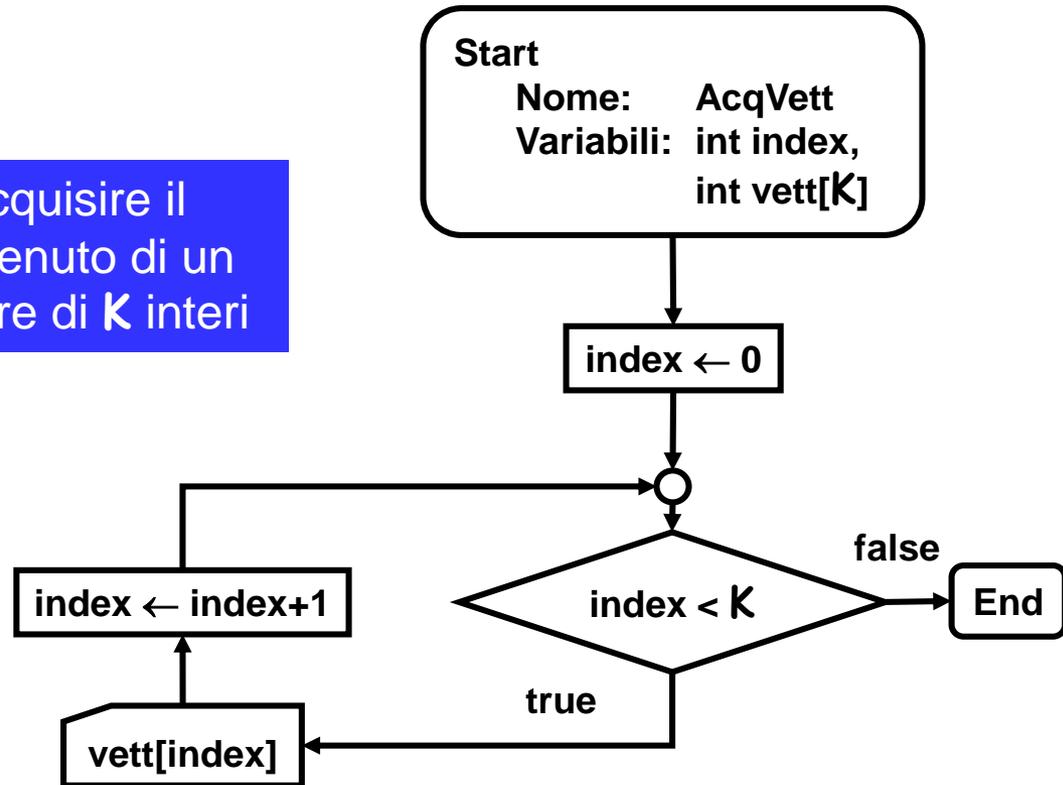
$$0 \leq \text{espressione a valore intero} \leq \text{dim}_{\text{Vettore}} - 1$$

nome_{Vettore} [indice]

Vettori

- **Esempio**

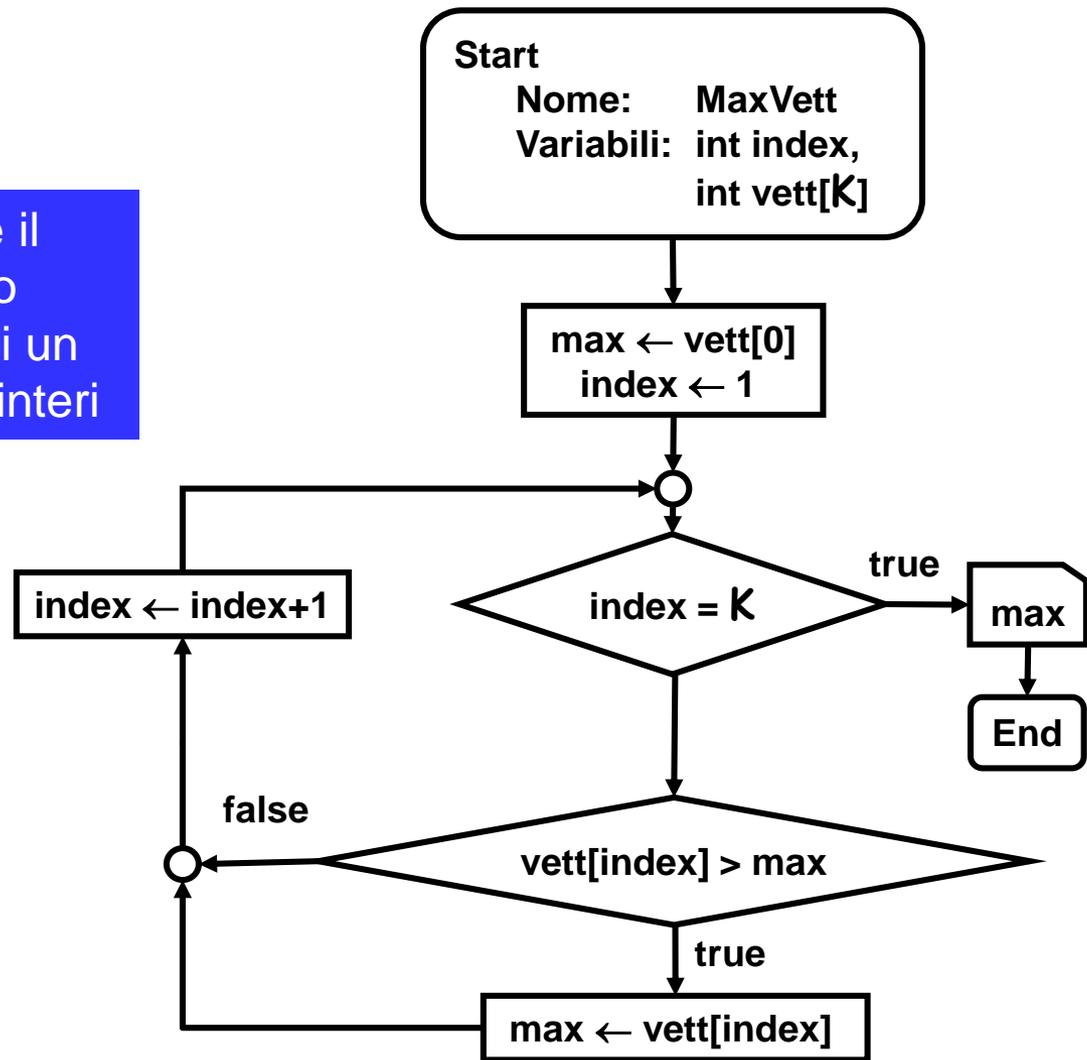
Acquisire il
contenuto di un
vettore di **K** interi



Vettori

- **Esempio**

Calcolare il
massimo
elemento di un
vettore di K interi



Matrici

- **Matrice di n x m elementi:**

definisce una corrispondenza biunivoca tra un multiinsieme omogeneo di n x m elementi e l'insieme di coppie di interi $\{(0,0), (0,1), \dots, (n-1, m-1)\}$

- **Esempio:**

Matrice di 5 x 2 interi

(0,0)	8	4	(0,1)
(1,0)	7	-1	(1,1)
(2,0)	15	12	(2,1)
(3,0)	4	-9	(3,1)
(4,0)	7	4	(4,1)

- **Definizione**

costanti intere

tipo_{Matrice} **nome**_{Matrice} [**dim**_{Righe}] [**dim**_{Colonne}]

Matrici

- **Accesso all'elemento di una matrice**

$0 \leq \text{espressione a valore intero} \leq \text{dim}_{\text{Colonne}} - 1$

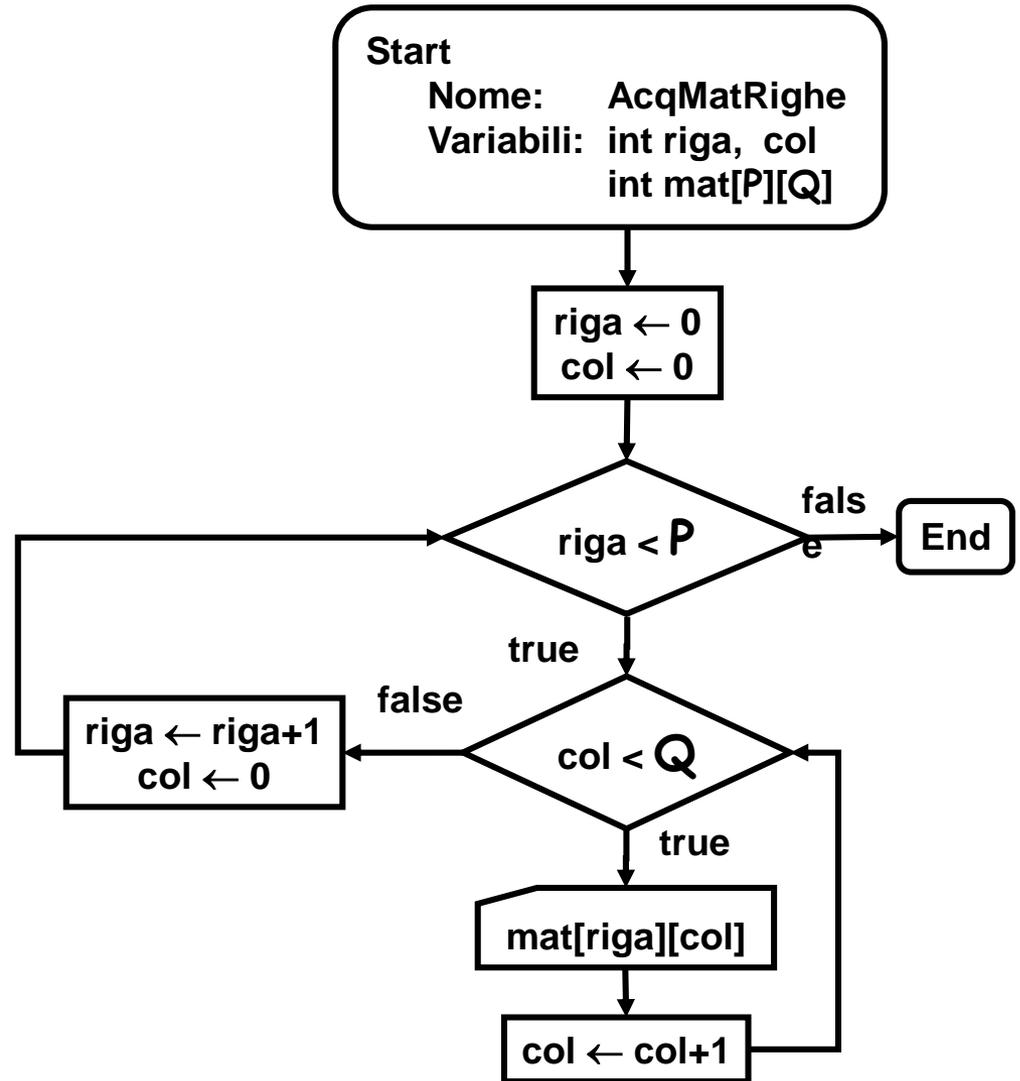
nome_{Matrice} [indice_{Riga}][indice_{Colonna}]

$0 \leq \text{espressione a valore intero} \leq \text{dim}_{\text{Righe}} - 1$

Matrici

- **Esempio**

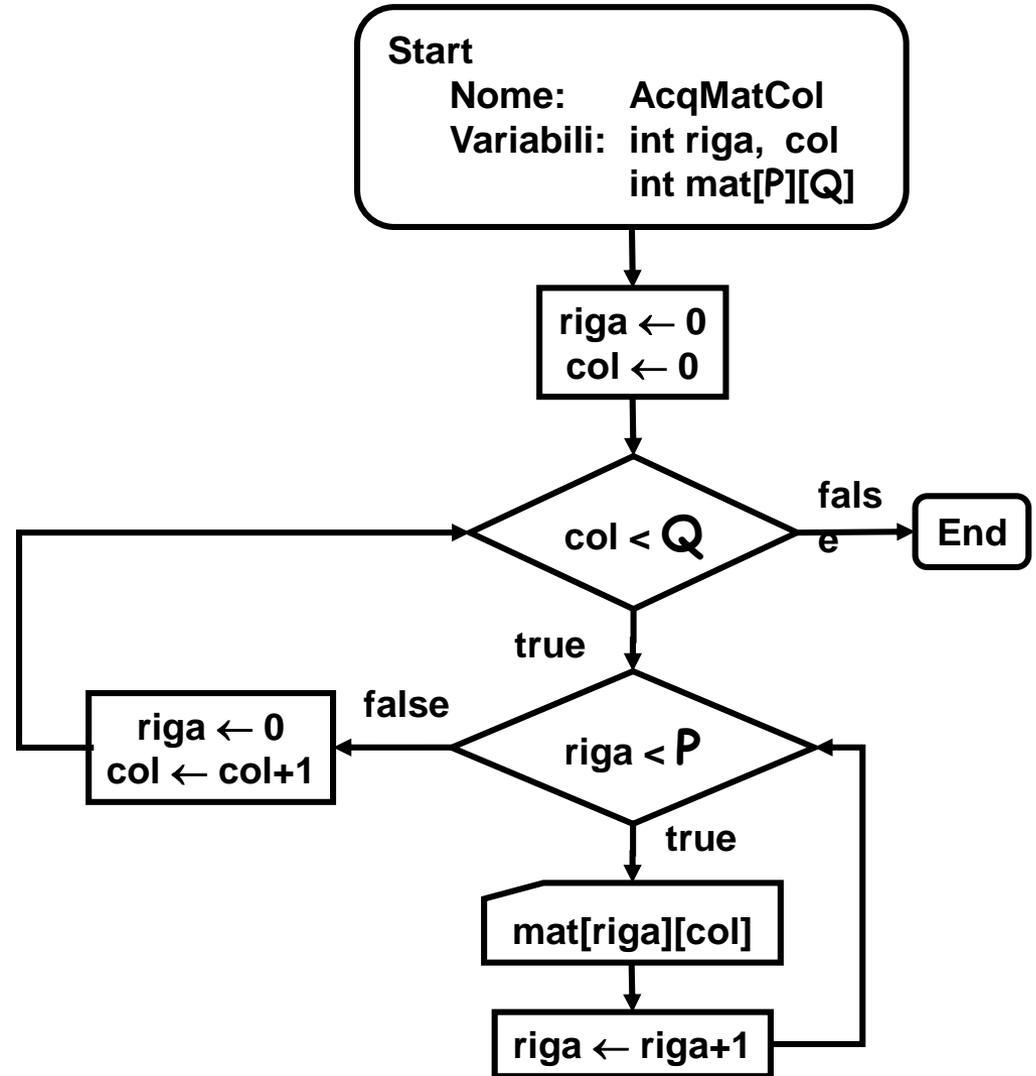
Acquisire il contenuto di una matrice di $P \times Q$ interi (per righe)



Matrici

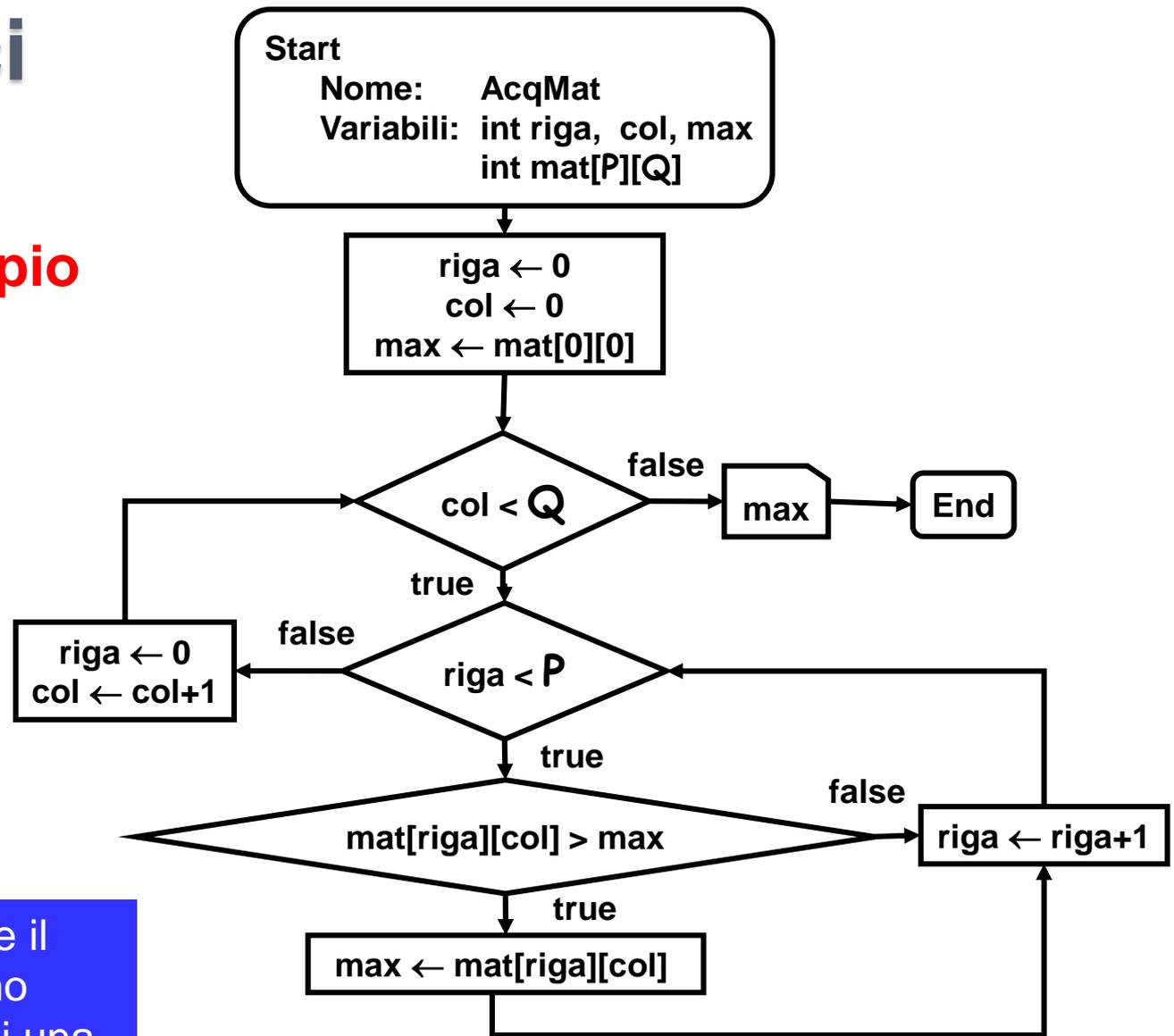
- **Esempio**

Acquisire il contenuto di una matrice di $P \times Q$ interi (per colonne)



Matrici

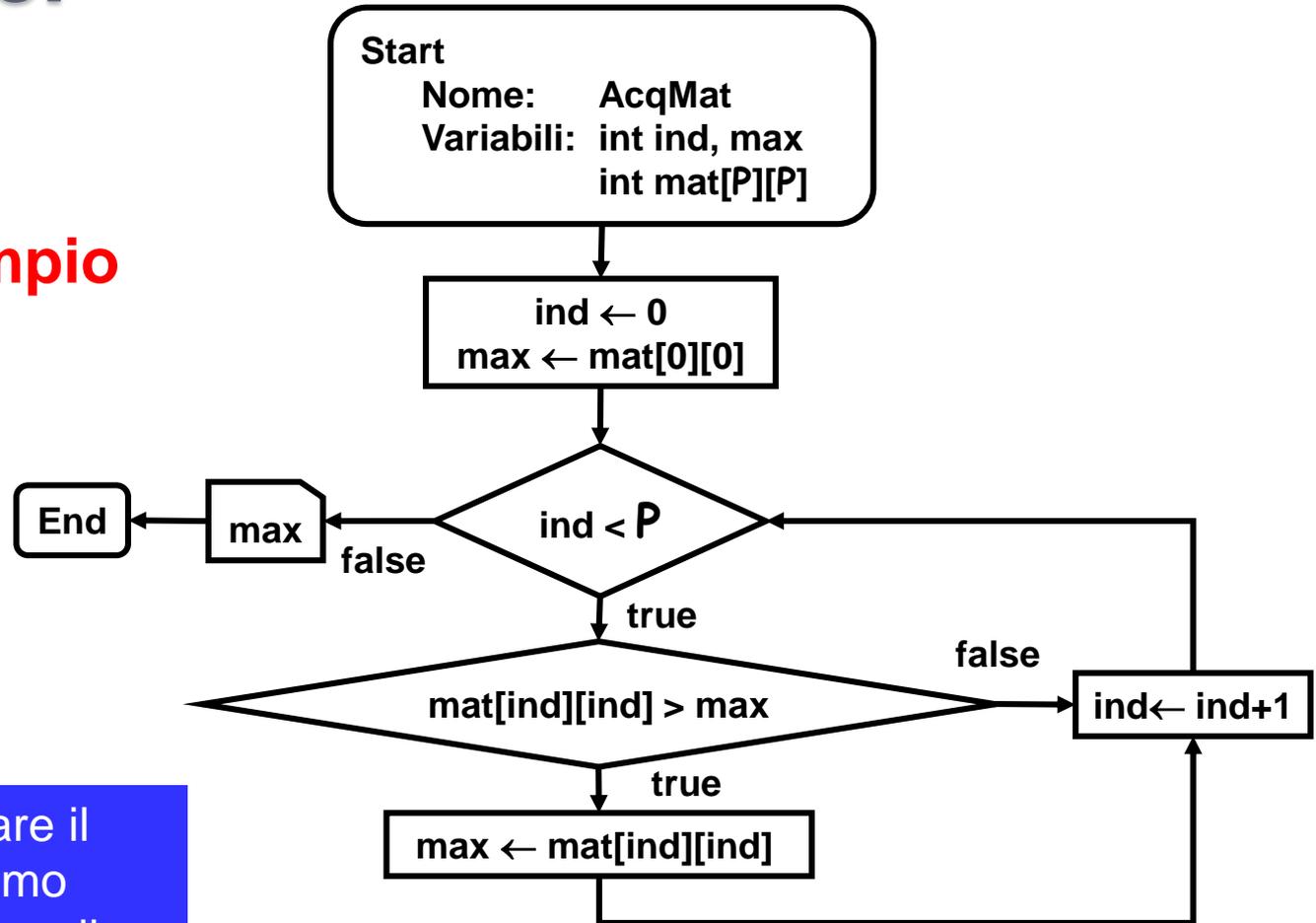
- **Esempio**



Calcolare il
massimo
elemento di una
matrice di P x Q

Matrici

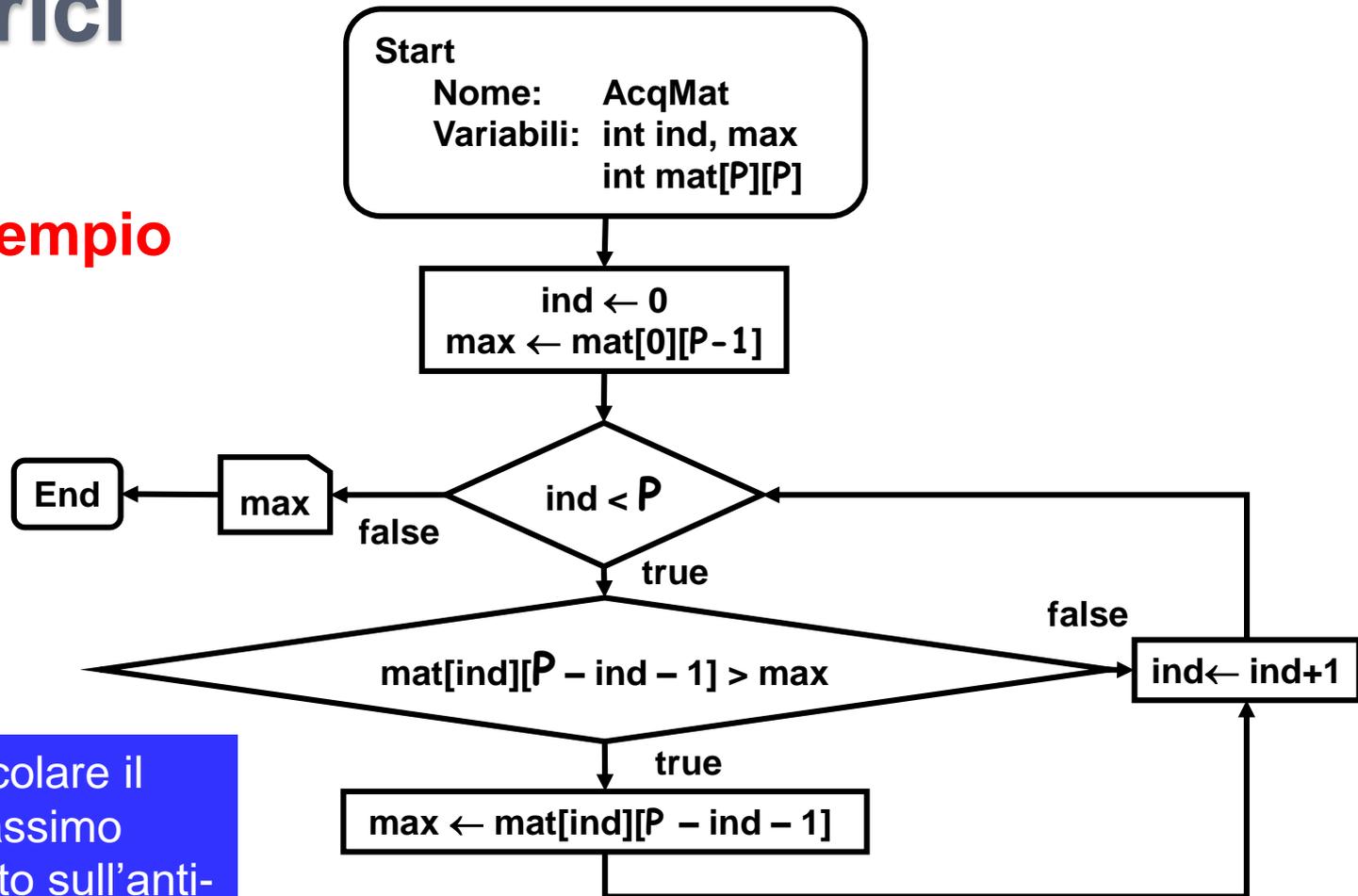
- **Esempio**



Calcolare il
massimo
elemento sulla
diagonale
principale di una
matrice di $P \times P$

Matrici

- **Esempio**



Calcolare il massimo elemento sull'anti-diagonale di una matrice di $P \times P$

Matrici

- **Prodotto tra matrici:**

siano date una matrice **A** di dimensione **m x n**
ed una seconda matrice **B** di dimensioni **n x p**.
Viene definito prodotto matriciale di **A** per **B** (**A x B**) la matrice **C**, di dimensioni **m x p**, i cui termini **c_{i,j}** sono calcolati come segue:

$$c_{i,j} = \sum_{r=1}^n a_{i,r} * b_{r,j}$$

Matrici

- Esempio

Calcolare il prodotto tra due matrici di interi, di dimensioni $P \times Q$ la prima, $Q \times R$ la seconda

