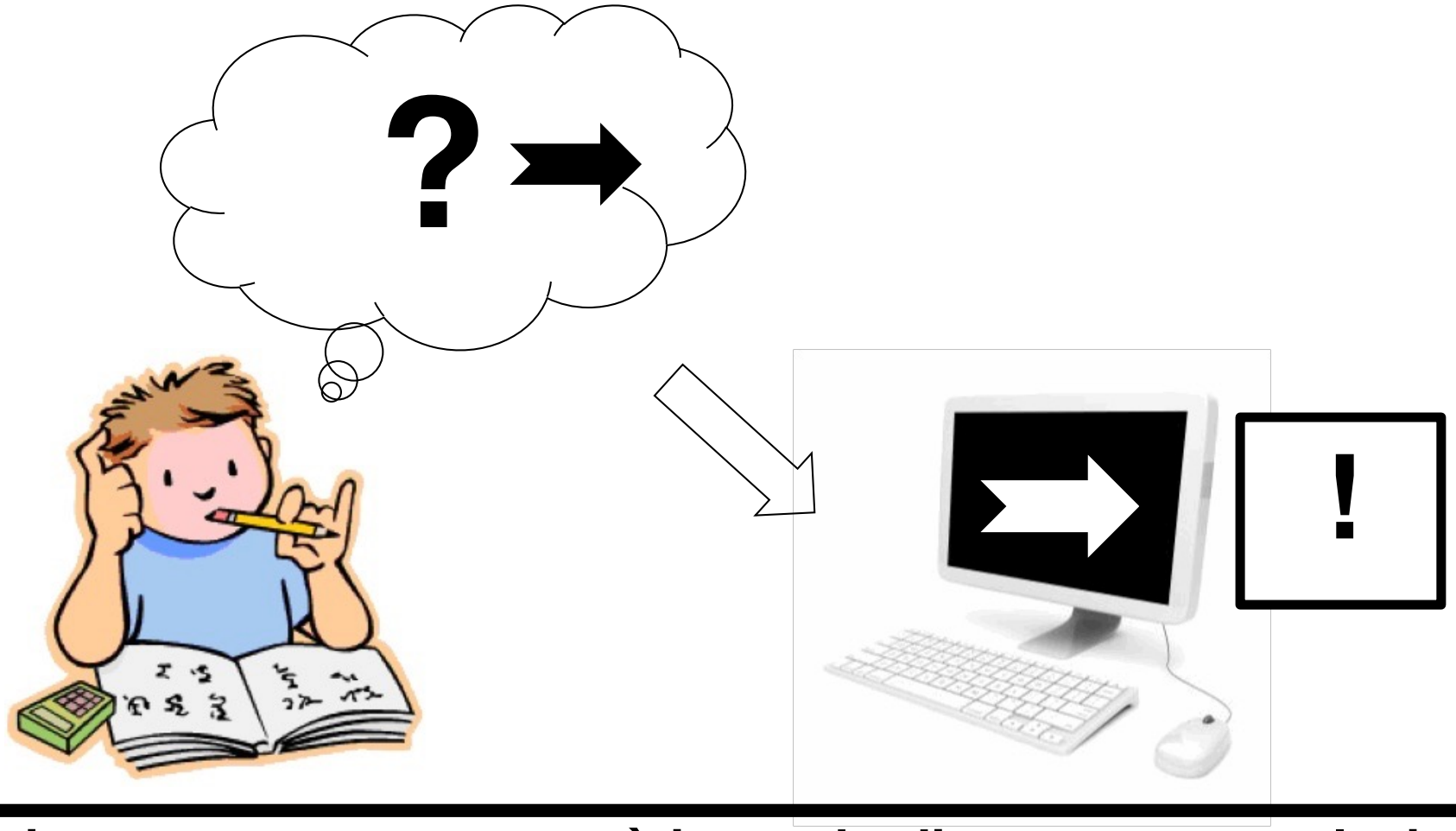
Principi e Metodi del Digitale
Lezione 5

Mario Verdicchio
Università degli Studi di Bergamo
A.A. 2026-2027

Esercizio

Fare il diagramma di flusso di un algoritmo che riceve in input due numeri e restituisce il loro massimo comune divisore.



Attenzione: un computer non è in grado di trovare una soluzione autonomamente. L'algoritmo deve prima essere ideato e successivamente trasferito nel computer sotto forma di programma, affinché venga eseguito e fornisca una soluzione.

Definizione

Fare il diagramma di flusso di un algoritmo che riceve in input due numeri e restituisce il loro massimo comune divisore.

Il più grande numero che divide entrambi i numeri.

Ragionamento

Il più grande numero che divide entrambi i numeri.

Visto che deve essere un divisore di entrambi, non può essere più grande del più piccolo dei due numeri.
Ad es. dati 12 e 20, il loro MCD potrà al massimo essere grande come 12, ma non di più (altrimenti non potrebbe essere un divisore di 12!).

Ragionamento

L'MCD non può essere più grande del più piccolo dei due numeri dati.

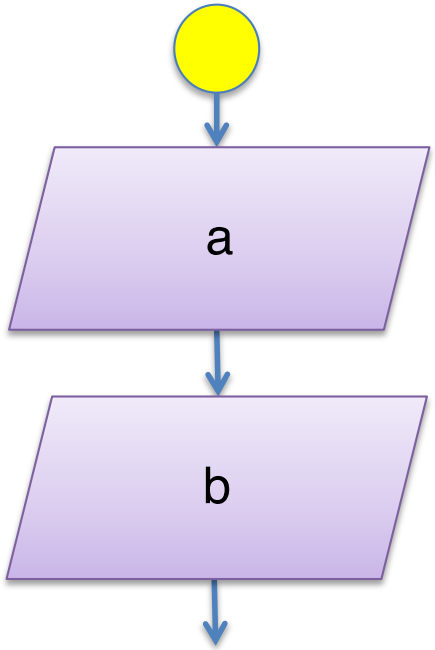
Per cercare, inizio col controllare se il più piccolo dei due li divide entrambi. Se è così, ho trovato il MCD. Altrimenti, provo con il numero immediatamente più piccolo (che ottengo sottraendo 1).

E così via…ripeto (andando a cercare numeri sempre più piccoli).

Rischio di andare avanti all'infinito? No, perché «alla peggio» il valore 1 li divide sicuramente entrambi.

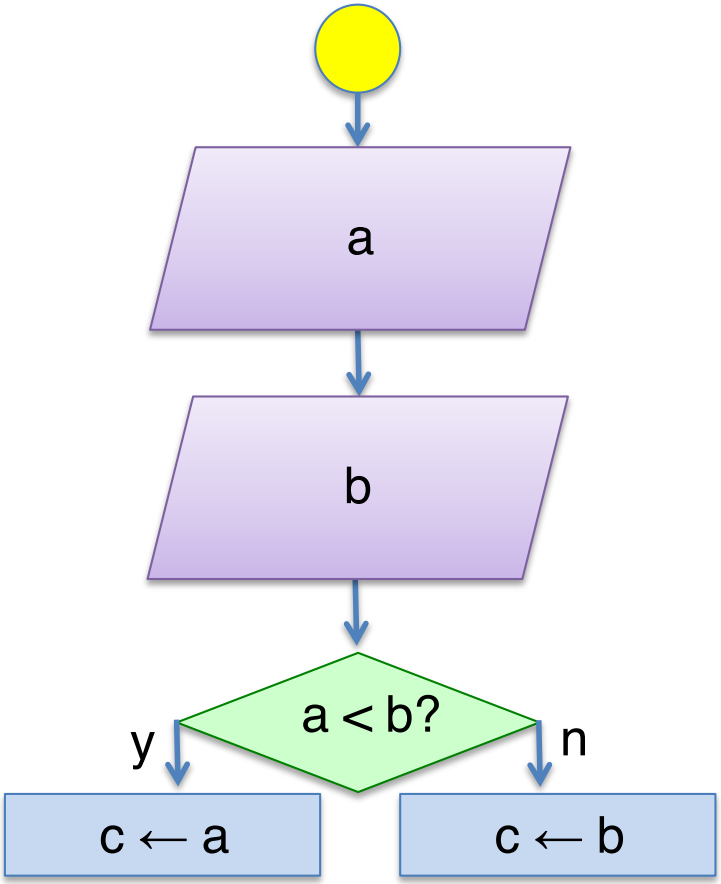
Costruzione dell'algoritmo

L'MCD non può essere più grande del più piccolo dei due numeri dati.



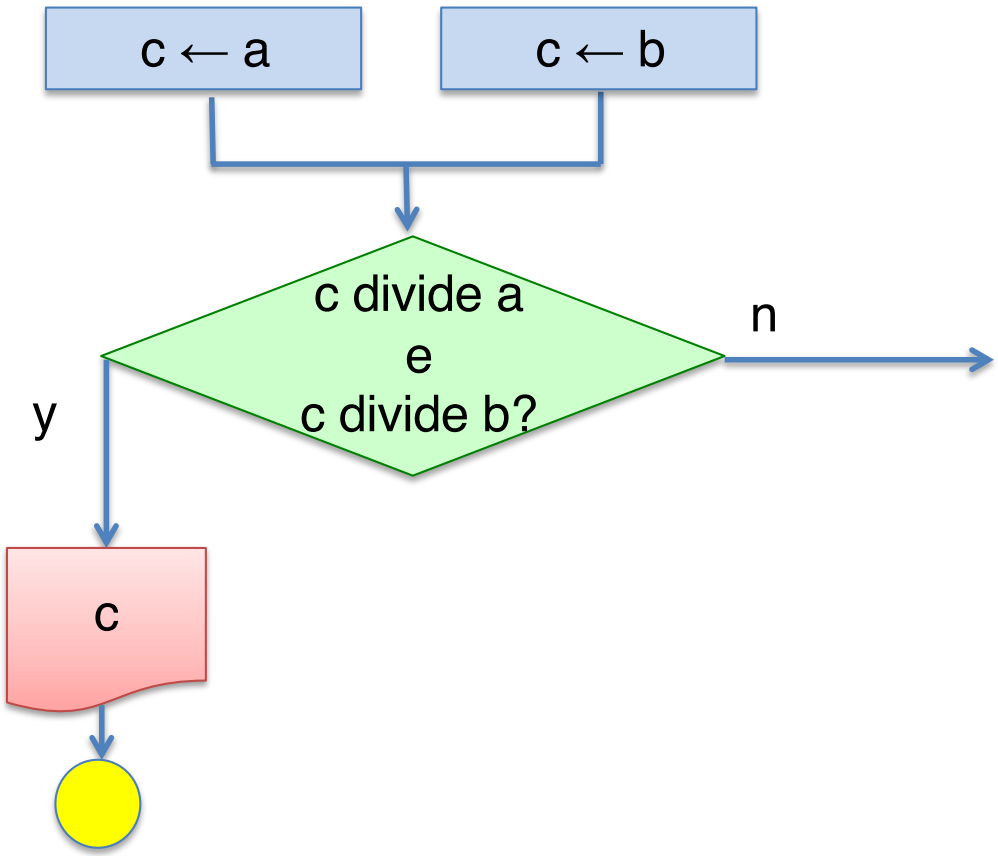
Costruzione dell'algoritmo

Per cercare, inizio col controllare se il più piccolo dei due li divide entrambi.



Costruzione dell'algoritmo

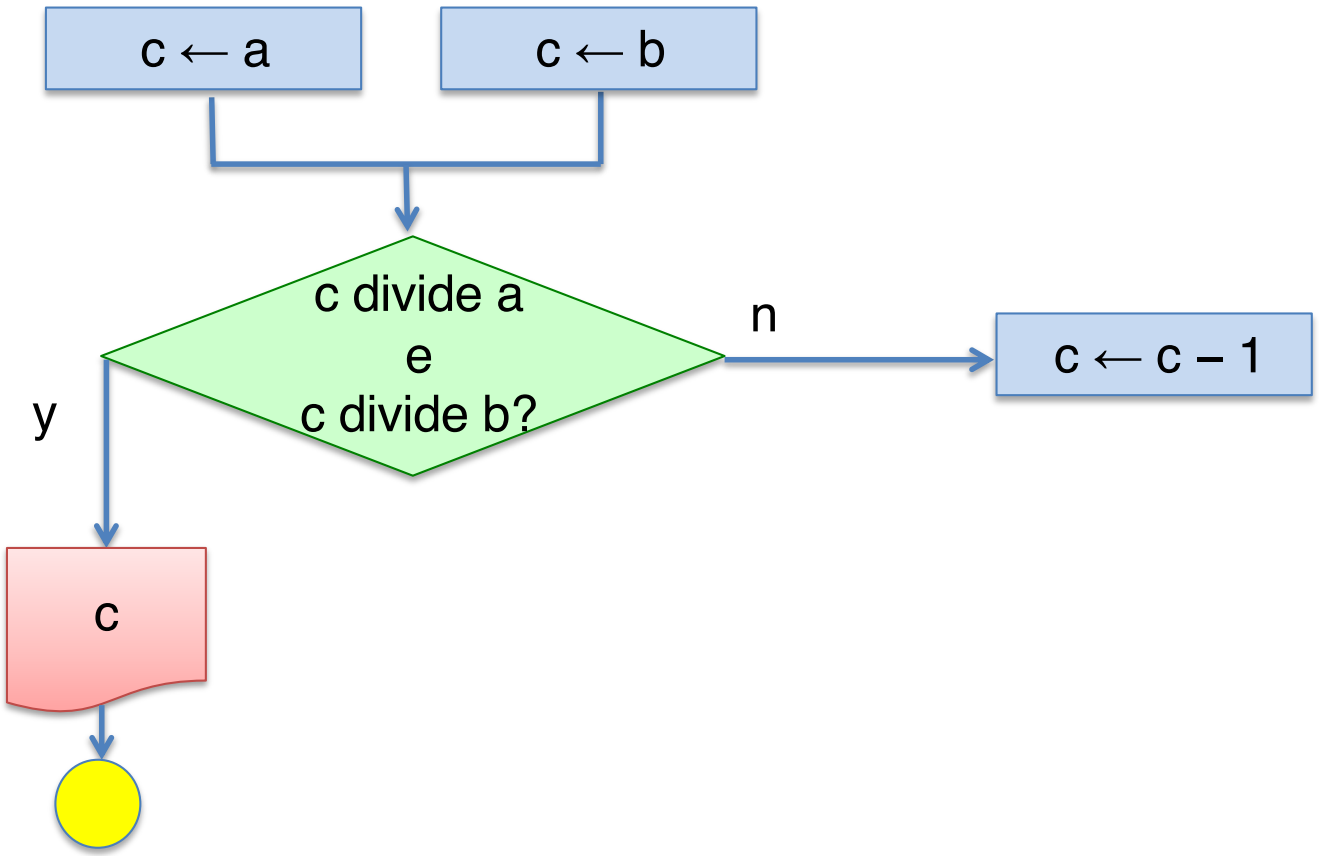
Per cercare, inizio col controllare se il più piccolo dei due li divide entrambi.



Il valore attuale di c non va bene perché non divide entrambi i numeri arrivati in input.
Dobbiamo provare con un nuovo valore (c diminuito di 1).
Ciò che va fatto, però, è esattamente quello che abbiamo fatto col vecchio valore di c; perciò, anziché riscrivere tutto, torniamo sopra con il flusso per ripetere le stesse azioni.
Non è solo questione di praticità: non sappiamo quante volte dovremo ripetere queste azioni al momento della scrittura dall'algoritmo (dipende dai valori di a e b); quindi non abbiamo informazioni sufficienti per determinare la lunghezza dell'esecuzione. Se costruiamo l'algoritmo con un ciclo, evitiamo il problema.

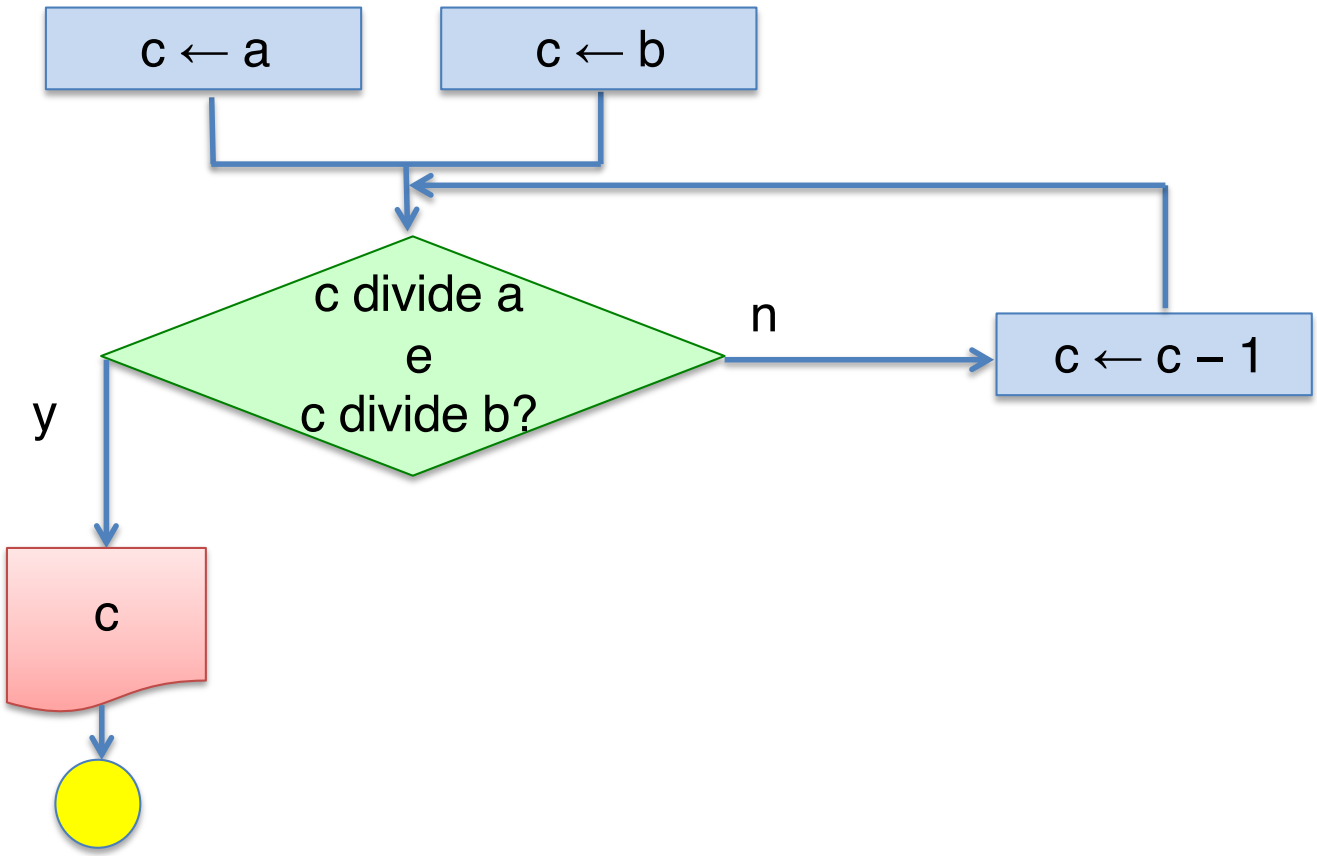
Costruzione dell'algoritmo

Altrimenti, provo con il numero immediatamente più piccolo (che ottengo sottraendo 1).

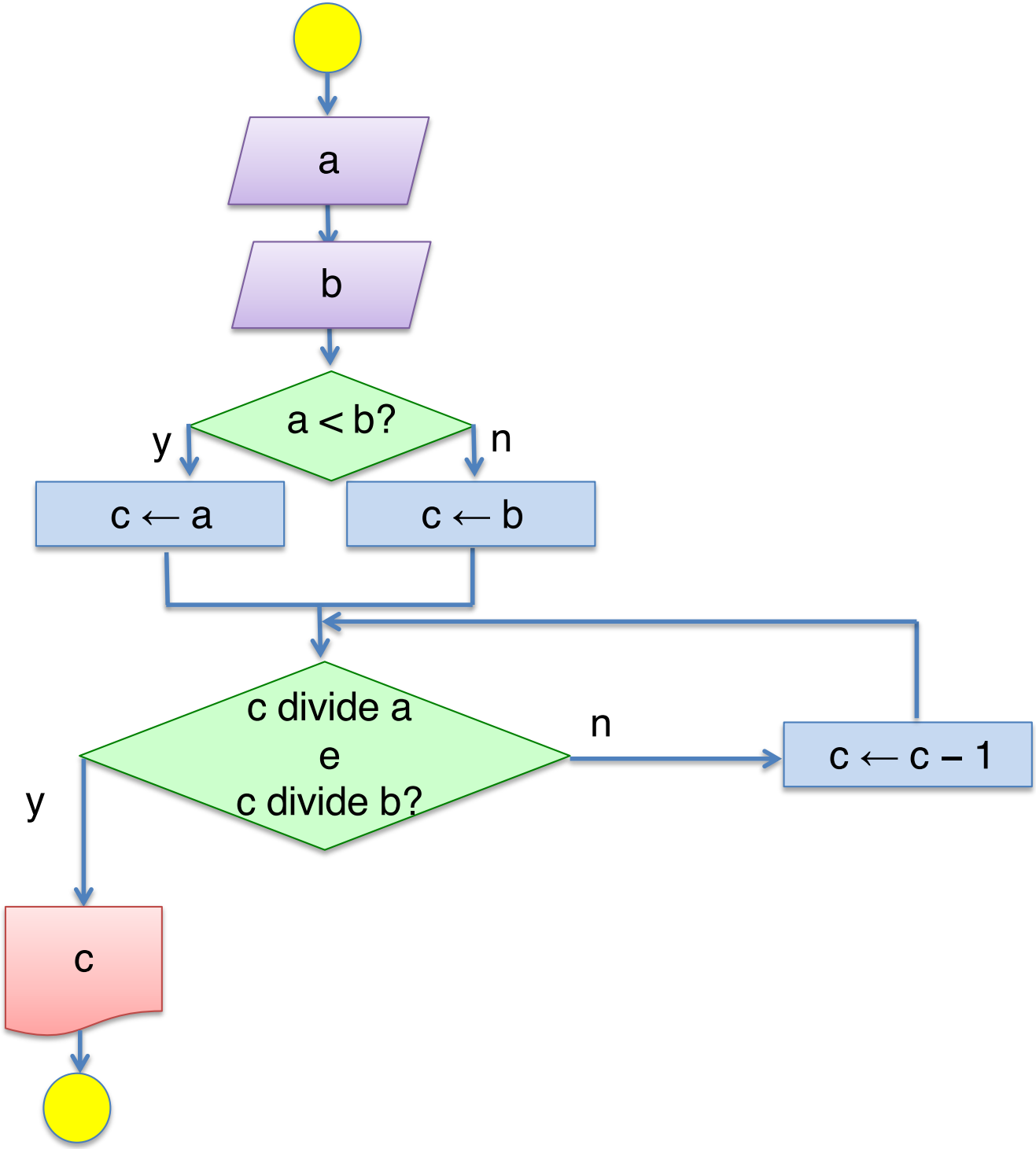


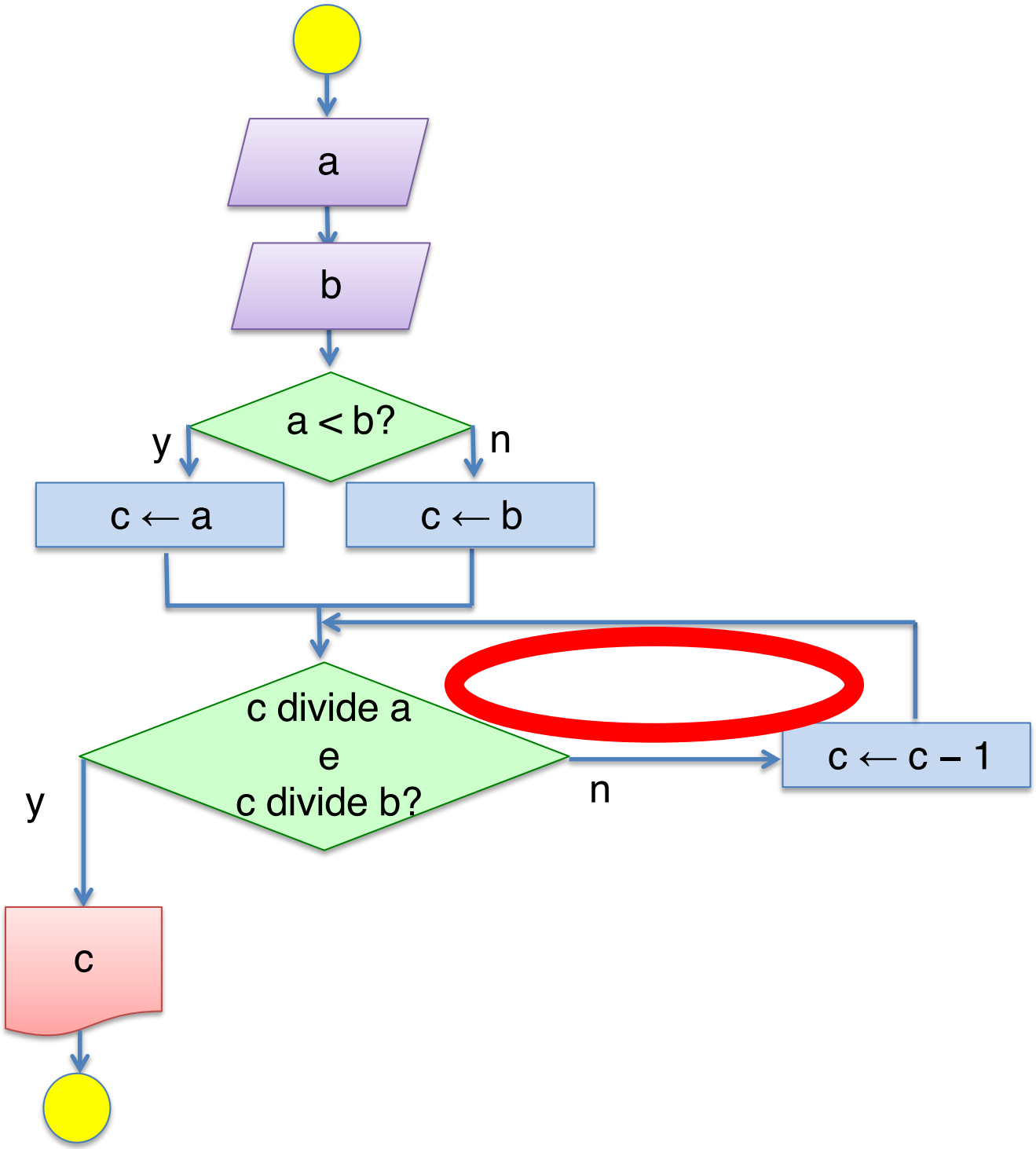
Costruzione dell'algoritmo

Altrimenti, provo con il numero immediatamente più piccolo (che ottengo sottraendo 1).



L'algoritmo

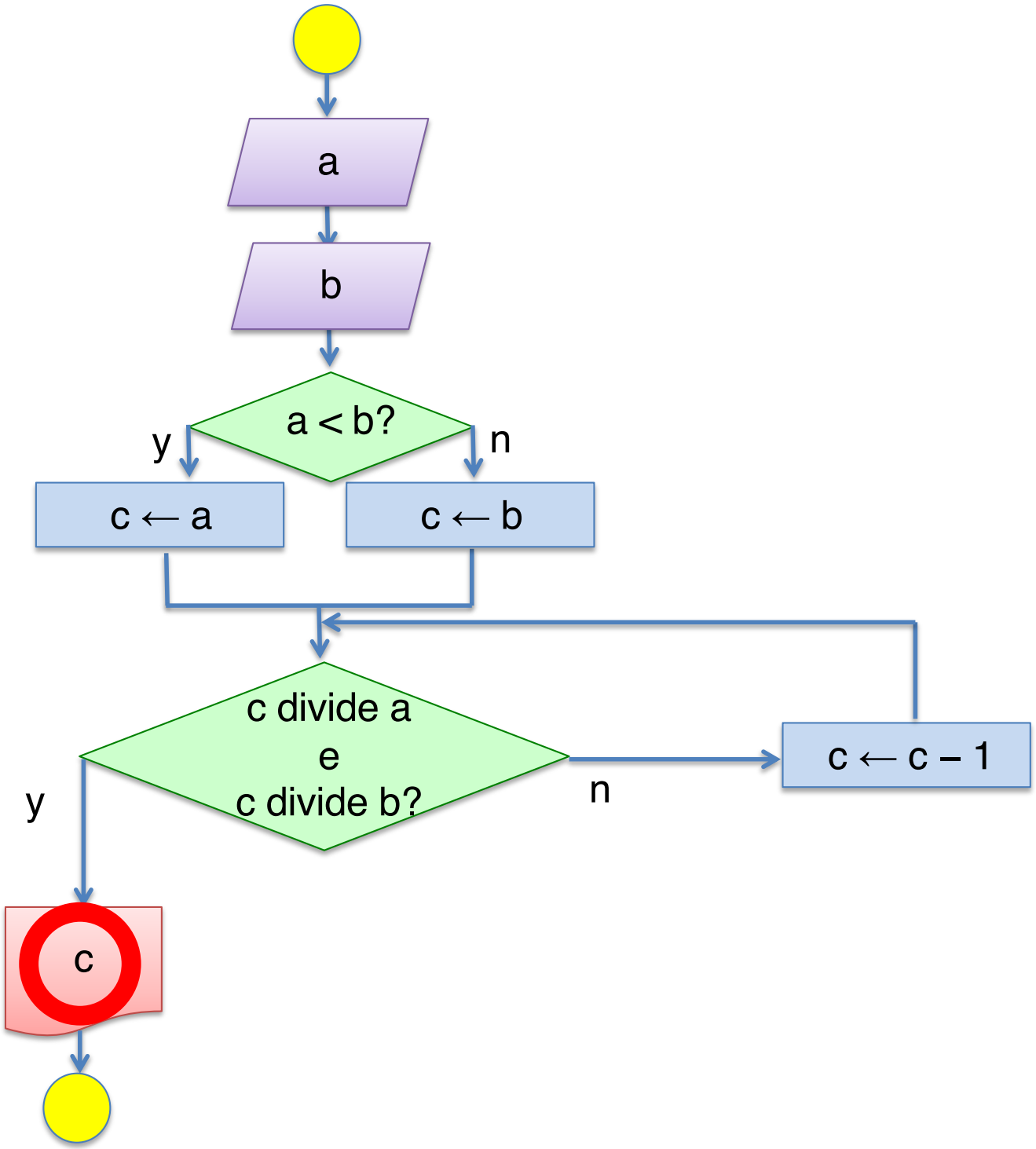




Dubbio 1:

Usciremo dal loop?

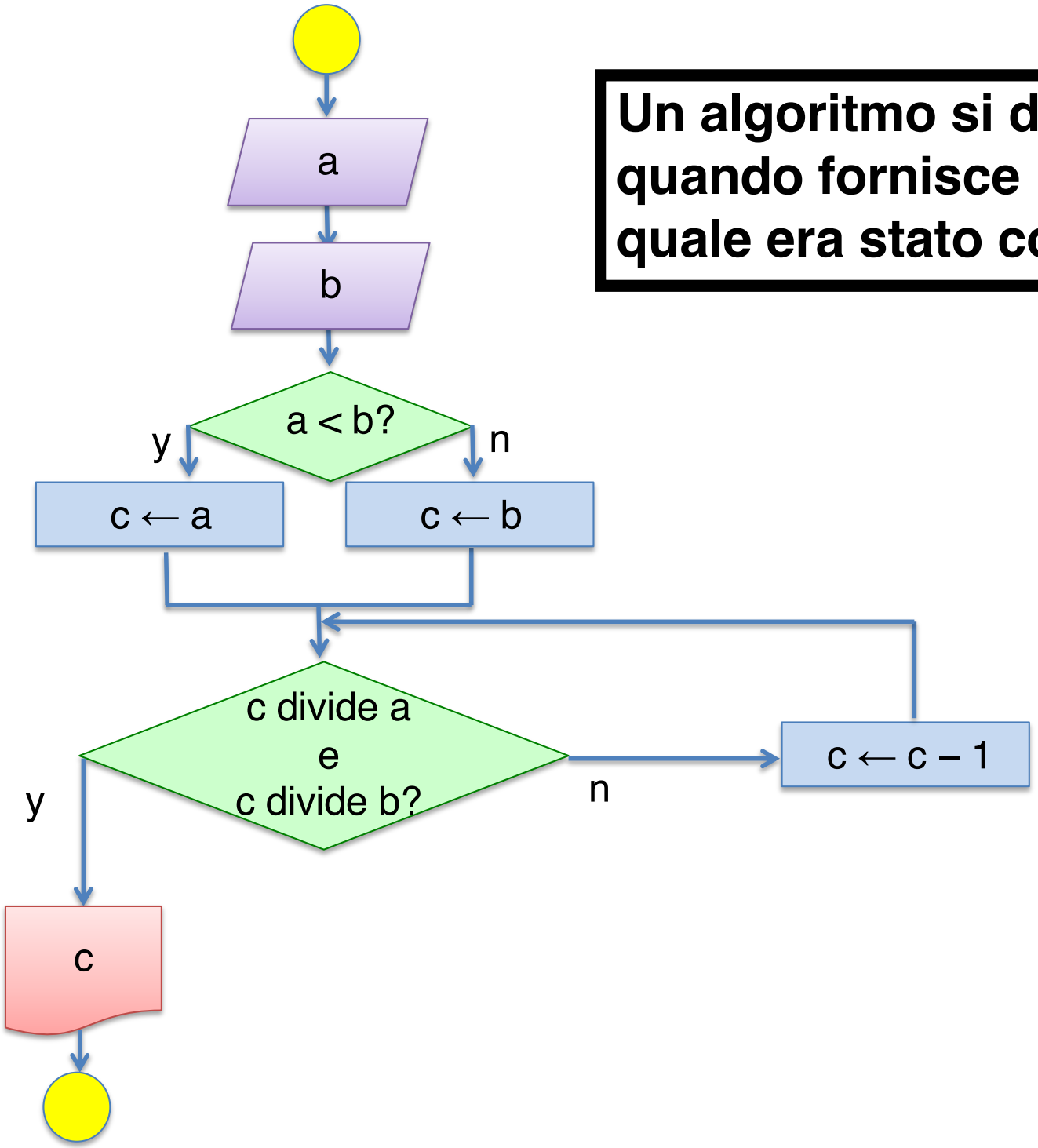




Dubbio 2:

c è davvero l'MCD?

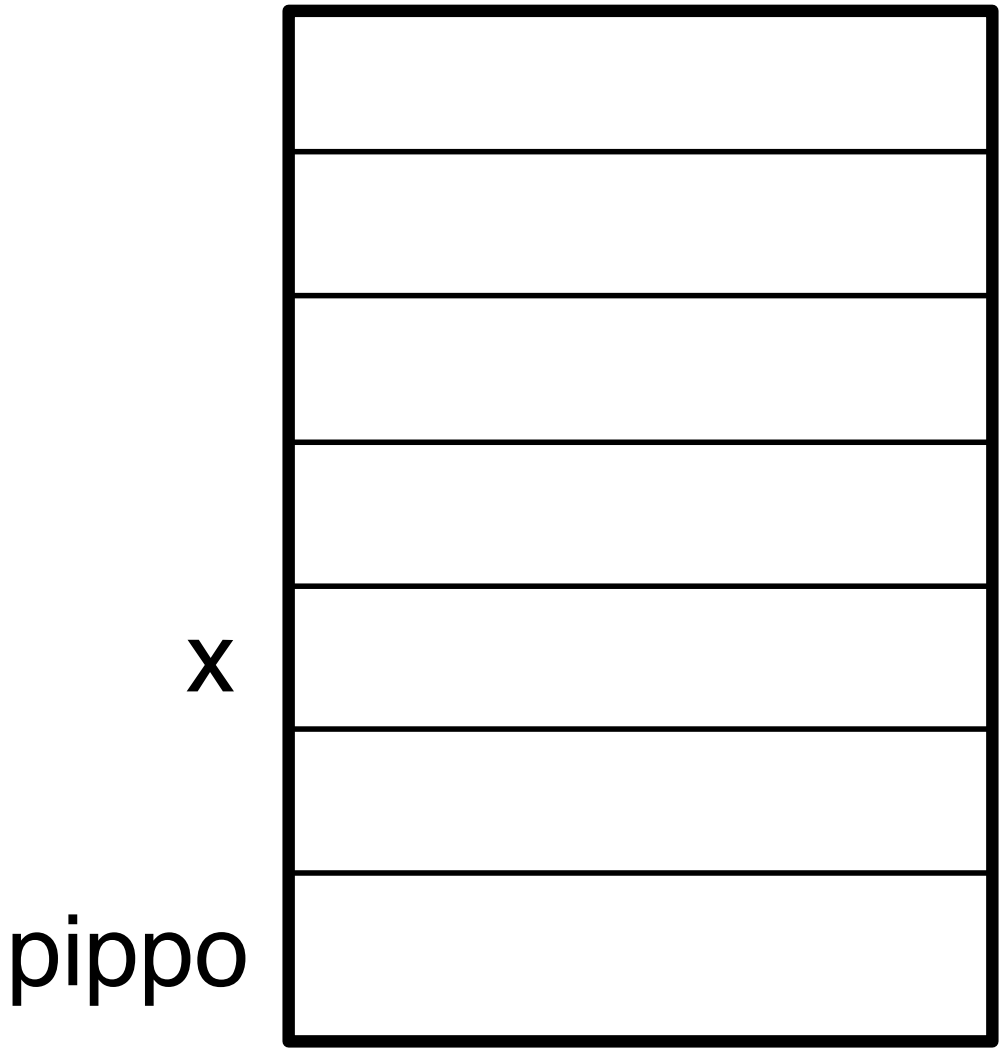




Un algoritmo si dice «corretto» quando fornisce il risultato per il quale era stato concepito.

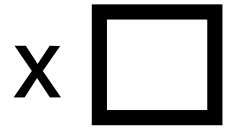


Variabili



Variabili

Astraendoci dalla struttura della memoria centrale, fino adesso abbiamo usato le variabili come contenitori di un singolo dato.

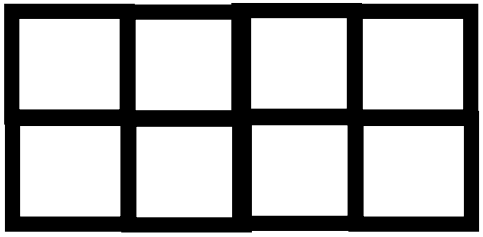




Variabili strutturate

Spesso, però, i dati sono organizzati sotto forma di liste (chiamate «vettori» o «array») e tabelle (chiamate «matrici»).

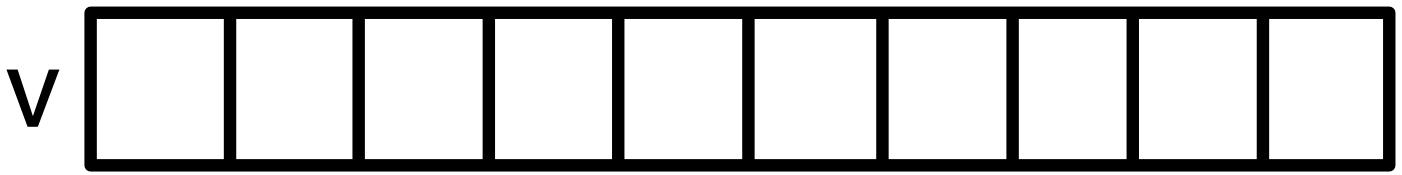
d 

paperino 

Uso dei vettori

1) La prima cosa da fare è dichiarare (con un'operazione) il nome di un vettore e la sua dimensione.

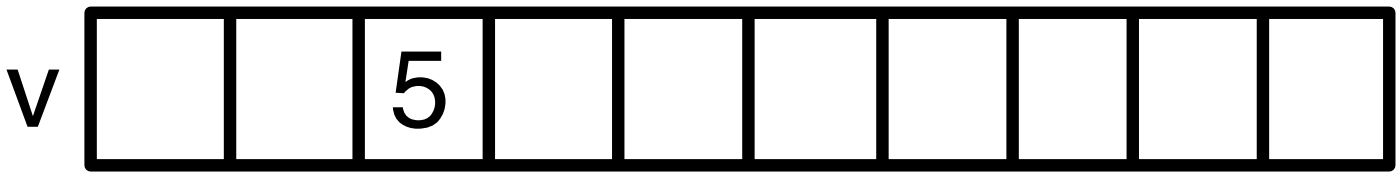
v[10]



Uso dei vettori

2) Dopo la dichiarazione, possiamo fare riferimento a qualunque «cella» del vettore, scrivendo il suo nome e l'indice che indica la cella che ci interessa.

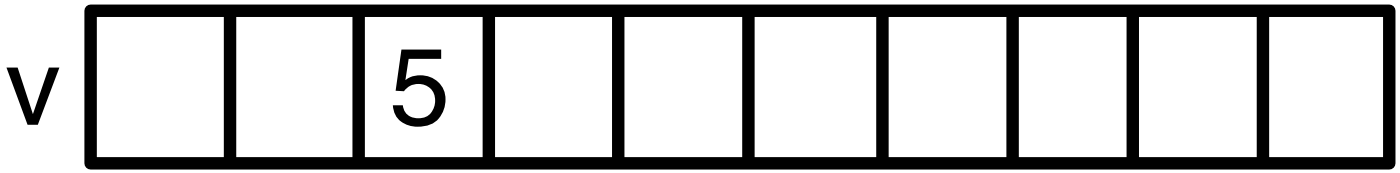
v[2] ← 5



Perché il 5 viene assegnato alla terza posizione?

v[2] ← 5

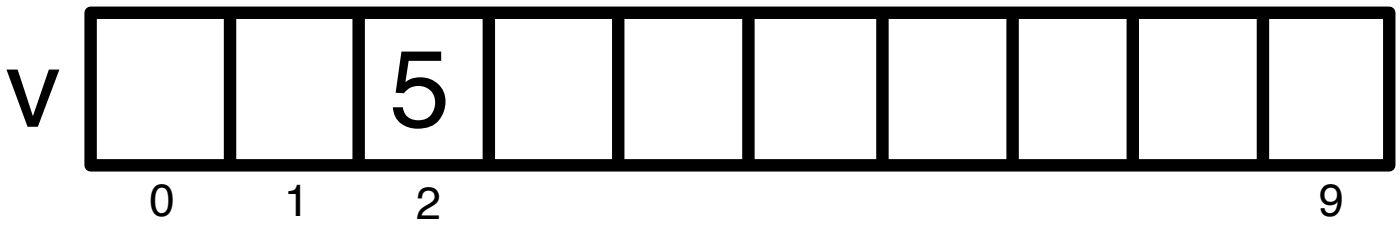




Perché gli indici delle celle iniziano dallo 0.

v[2] ← 5

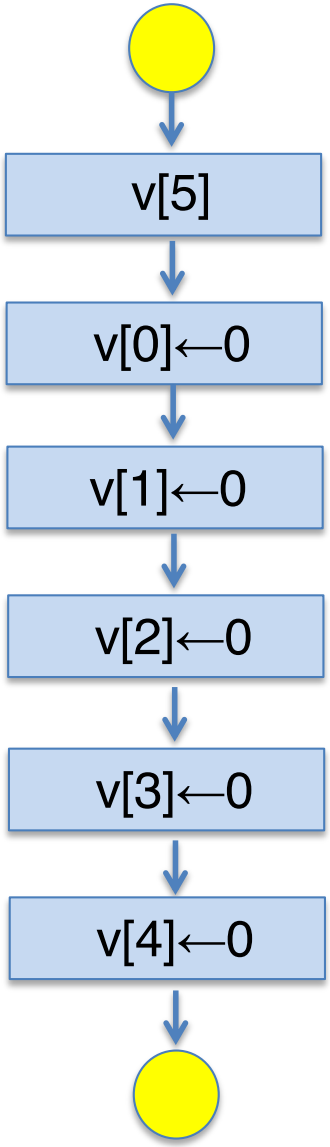




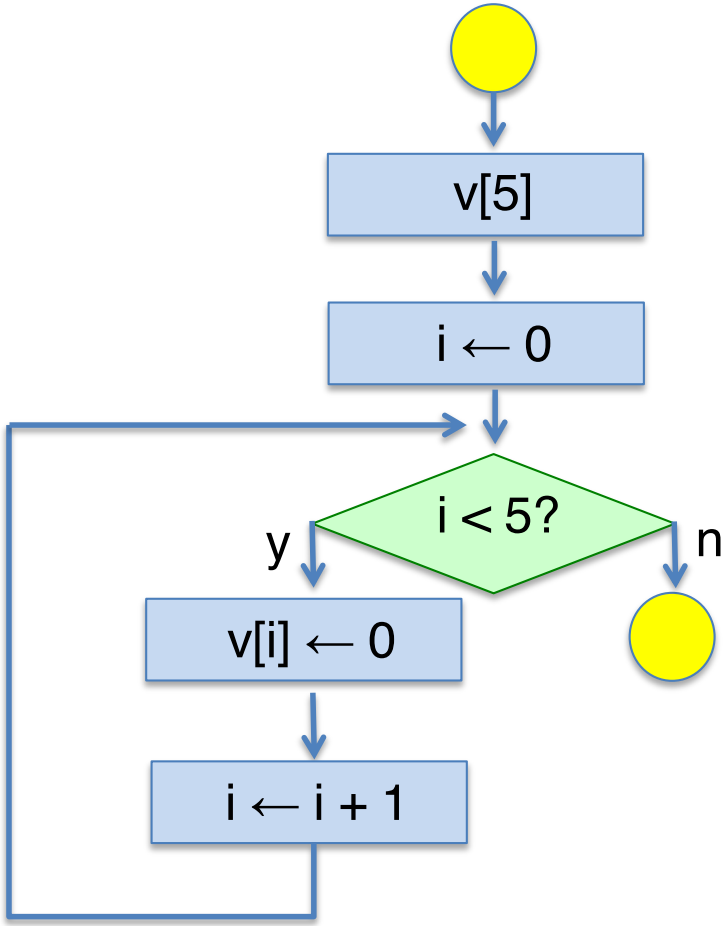
Esercizio

Fare il diagramma di flusso di un algoritmo che costruisce un vettore di dimensione 5 e lo riempie di zeri.

Soluzione corretta ma non parametrizzata.

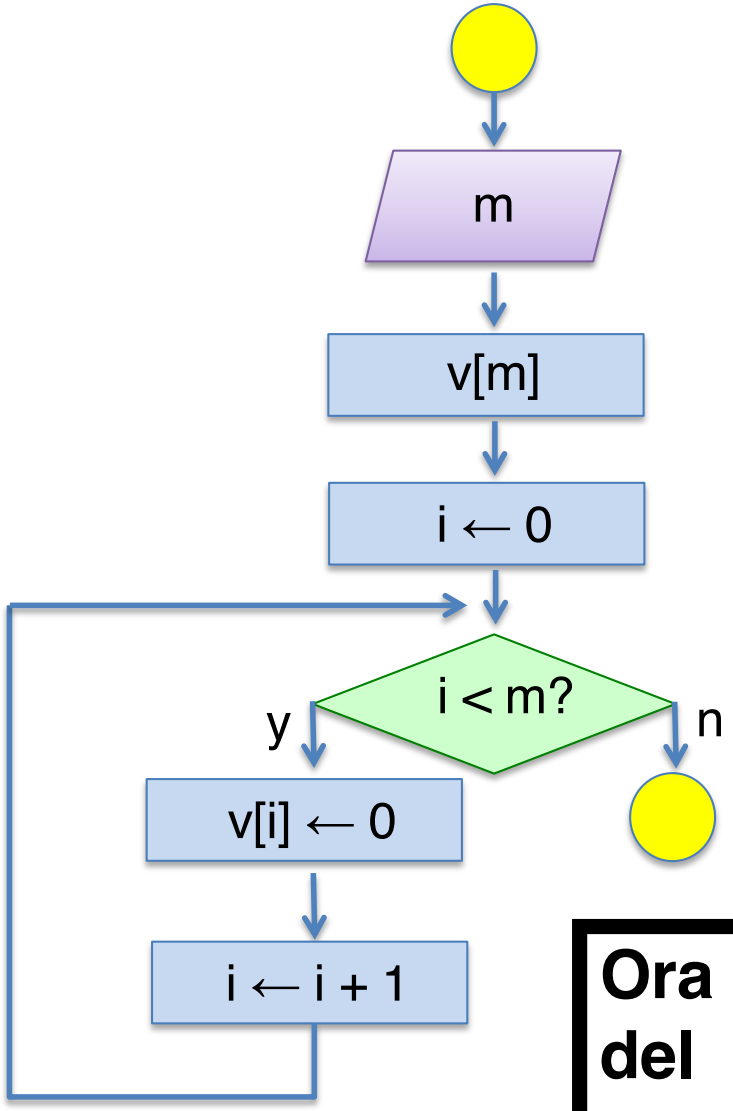


Soluzione corretta e parametrizzata.



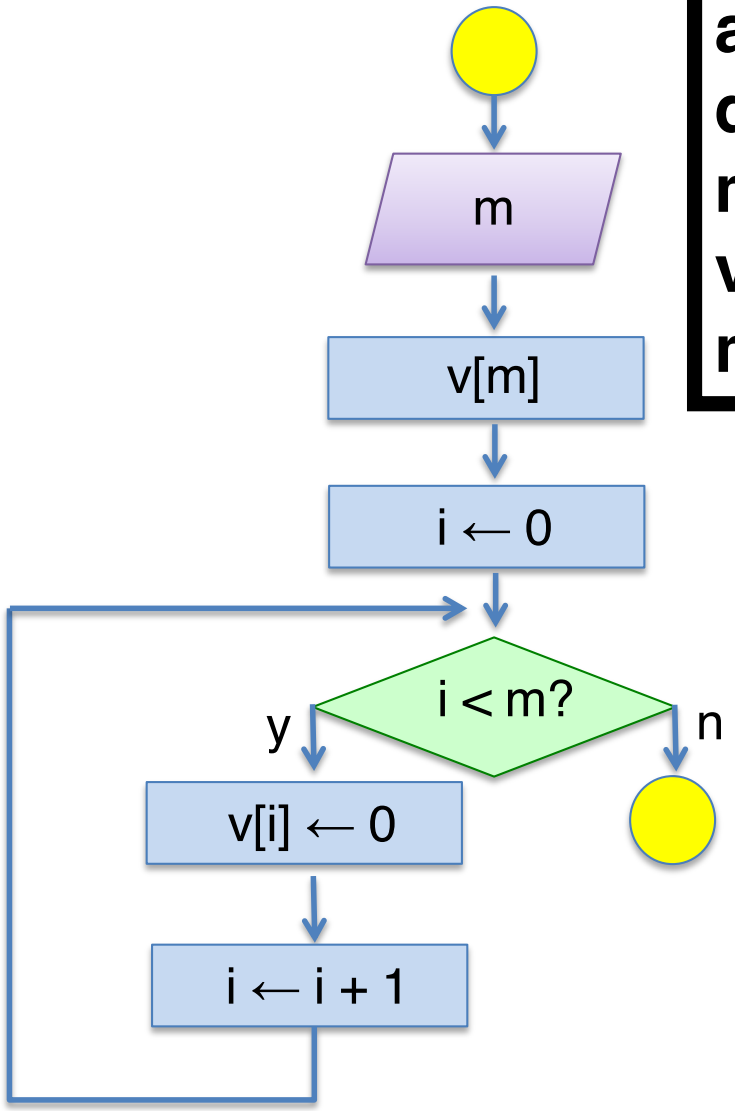
i è una variabile che funge da indice per le varie posizioni nel vettore.

Estendiamo la parametrizzazione.



Ora anche la dimensione del vettore è un parametro, e lo facciamo decidere all'utente.

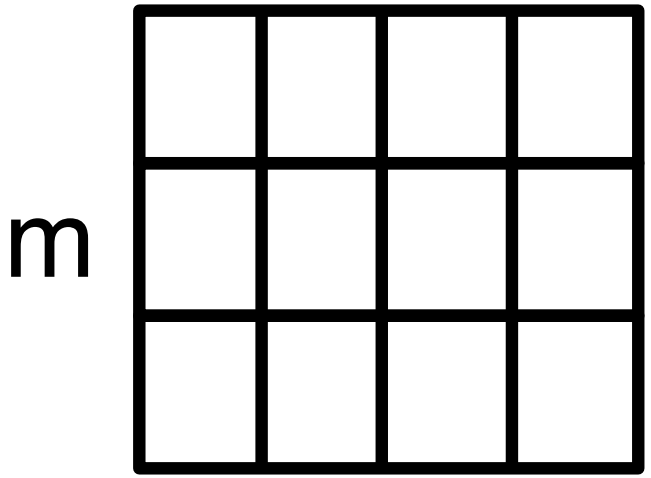
Questo è il diagramma di flusso di un algoritmo che riceve dall'utente un numero m, costruisce un vettore di dimensione m e lo riempie di zeri.



Uso delle matrici

1) Come prima, si inizia con la dichiarazione. In questo caso, le dimensioni da specificare sono 2: il numero di righe e il numero di colonne.

```
m[3][4]
```

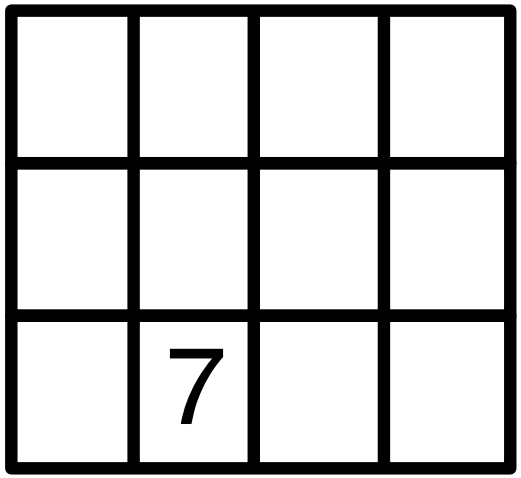


Uso delle matrici

2) Come prima, dopo la dichiarazione possiamo fare riferimento a qualunque «cella» della variabile strutturata, scrivendo il suo nome e i due indici (riga, colonna) che indicano la cella che ci interessa.

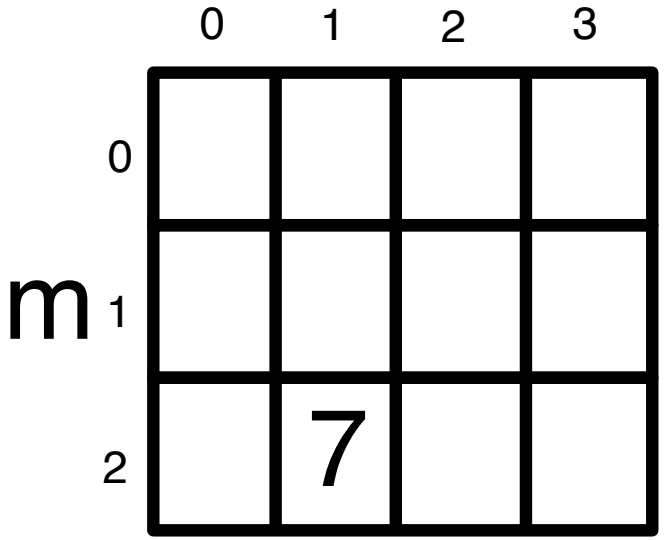
m[2][1]← 7

m



Non dimentichiamoci che gli indici iniziano sempre con lo 0.

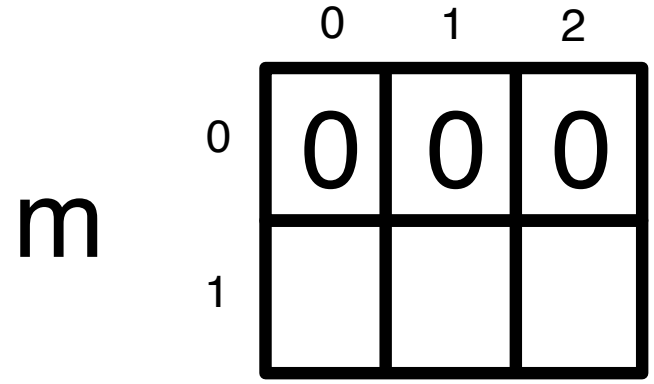
m[2][1]← 7



Esercizio

Fare il diagramma di flusso di un algoritmo che costruisce una matrice 2 x 3 e la riempie di zeri.

Idea di base



Costruiamo la matrice e la riempiamo riga per riga.

Usiamo una variabile i come indice della riga
Usiamo una variabile j come indice della colonna

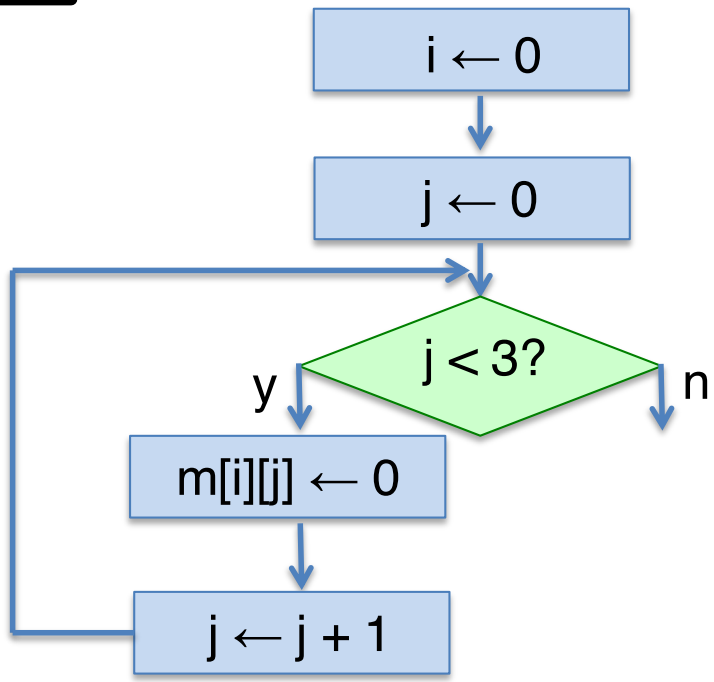
Per riempire la prima riga, i è fisso a 0, e j varia da 0 a 2.

Riga fissa, ciclo sull'indice delle colonne

m

	0	1	2
0	0	0	0
1			

Per riempire la prima riga, i è fisso a 0, e j varia da 0 a 2.

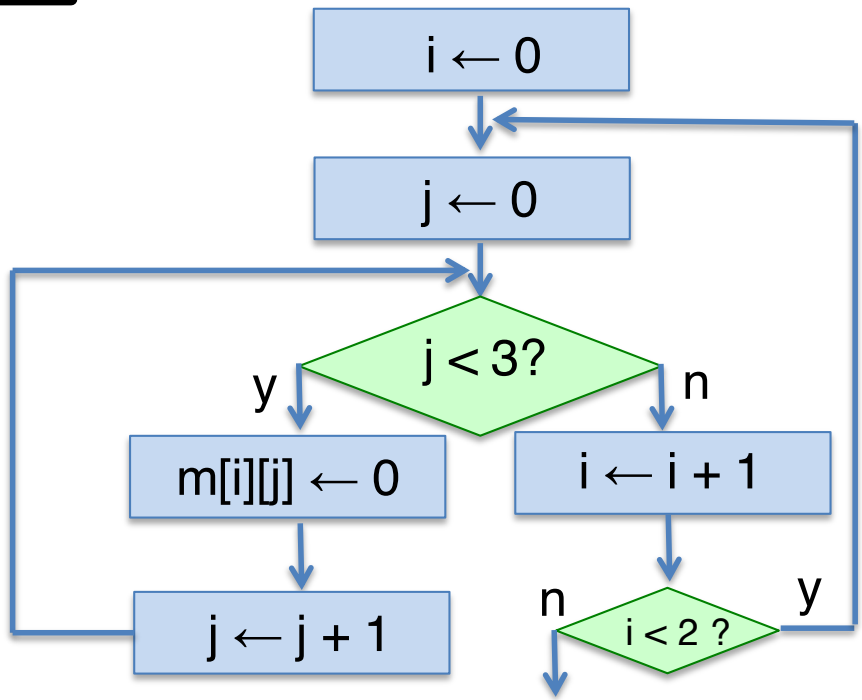


Ciclo sull'indice delle righe

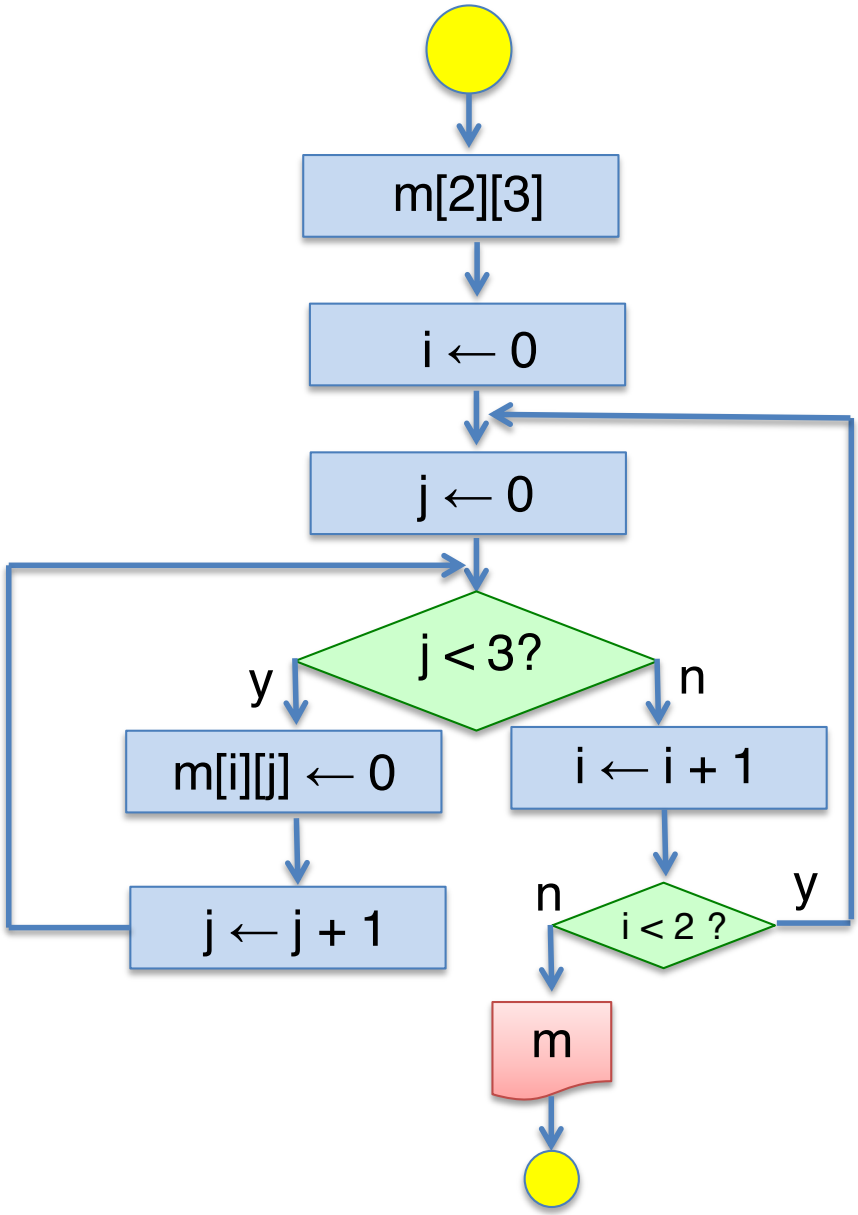
m

	0	1	2
0	0	0	0
1	0	0	0

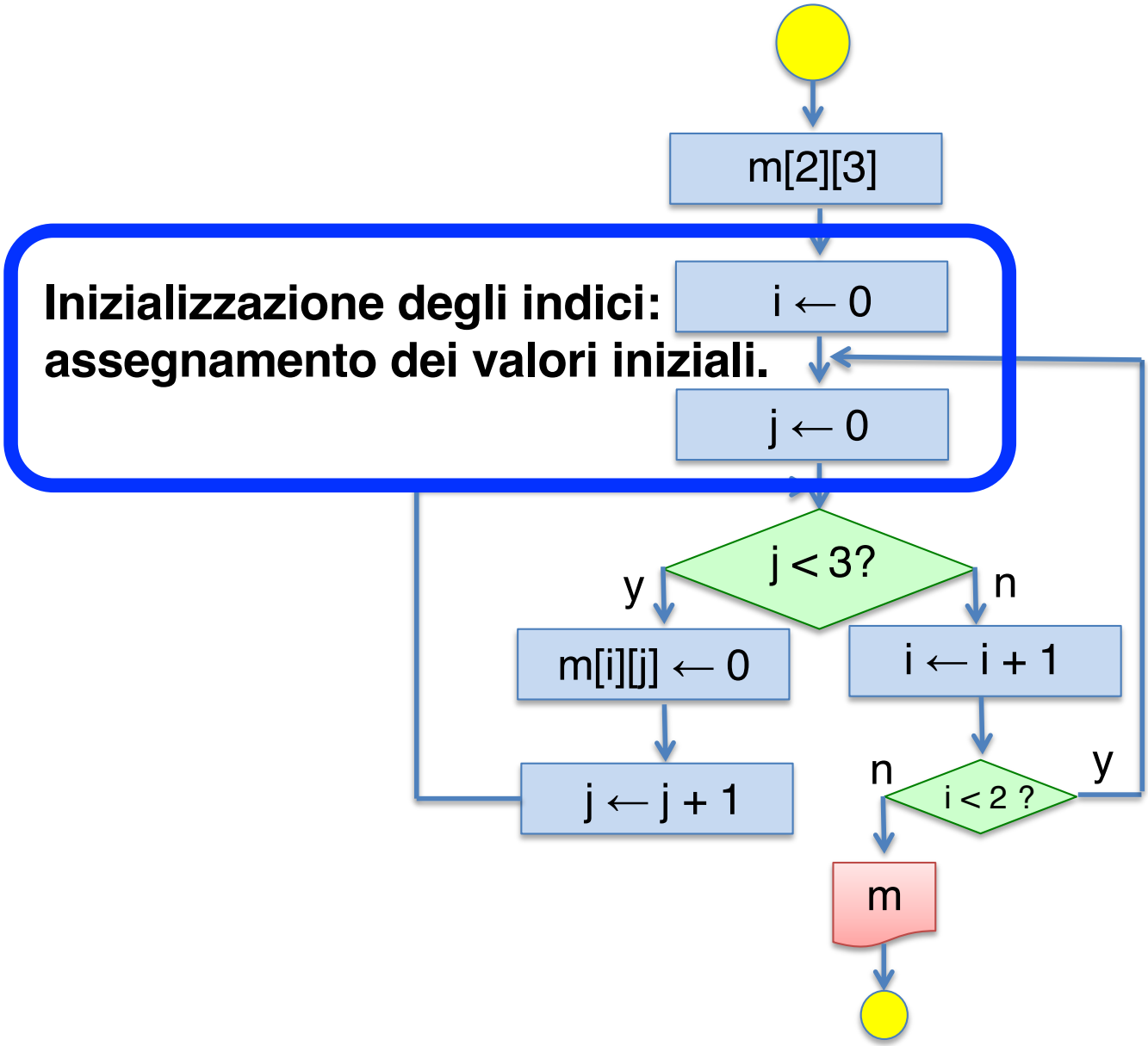
Quando la riga si completa (cioè l'indice colonna arriva al valore massimo), si passa alla riga successiva.



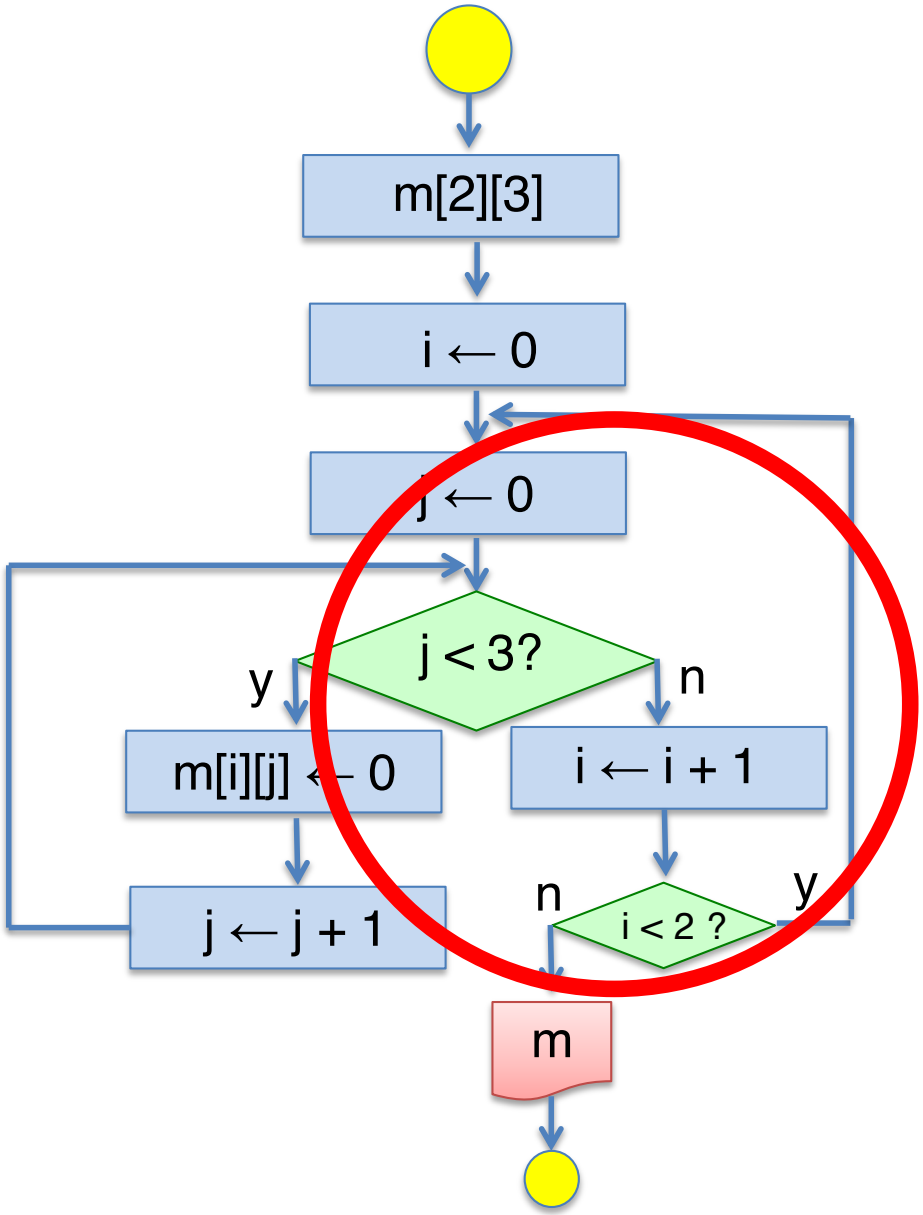
Algoritmo completo



Algoritmo completo



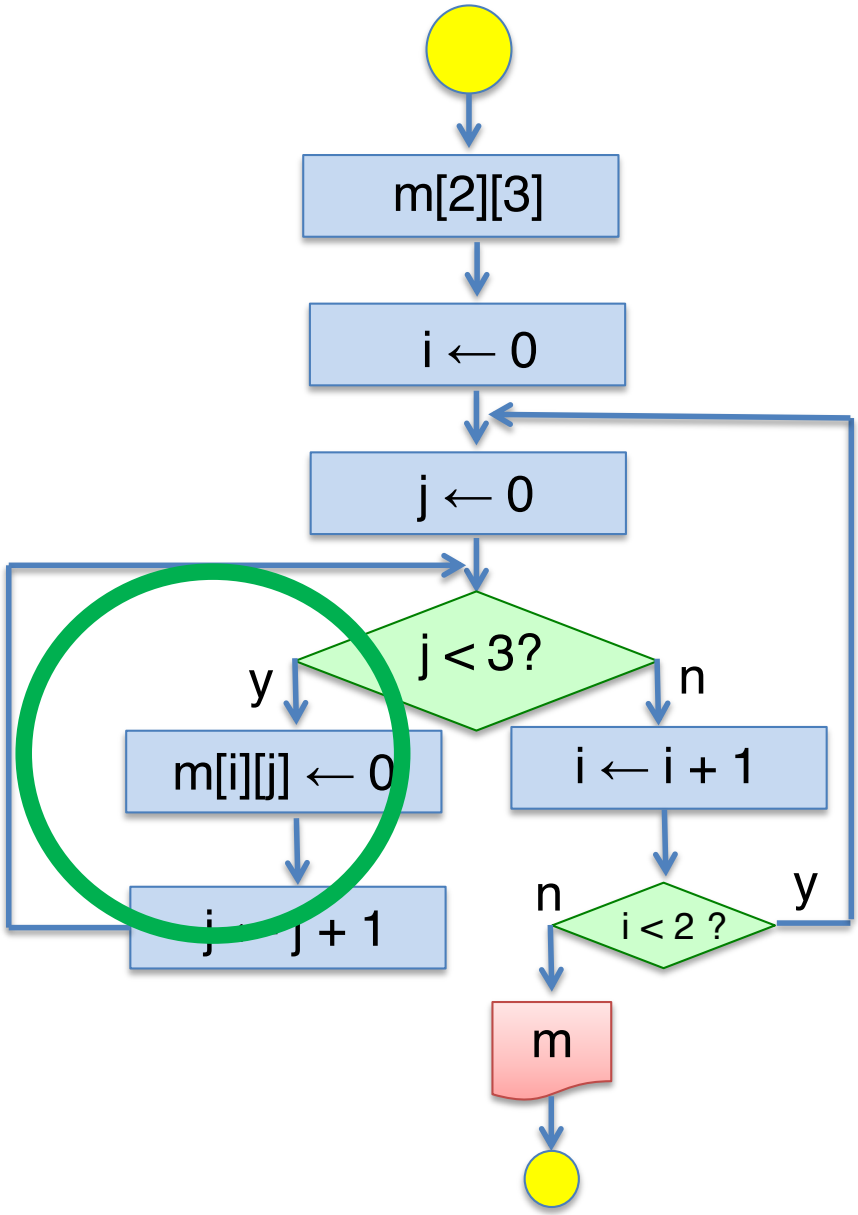
Algoritmo completo



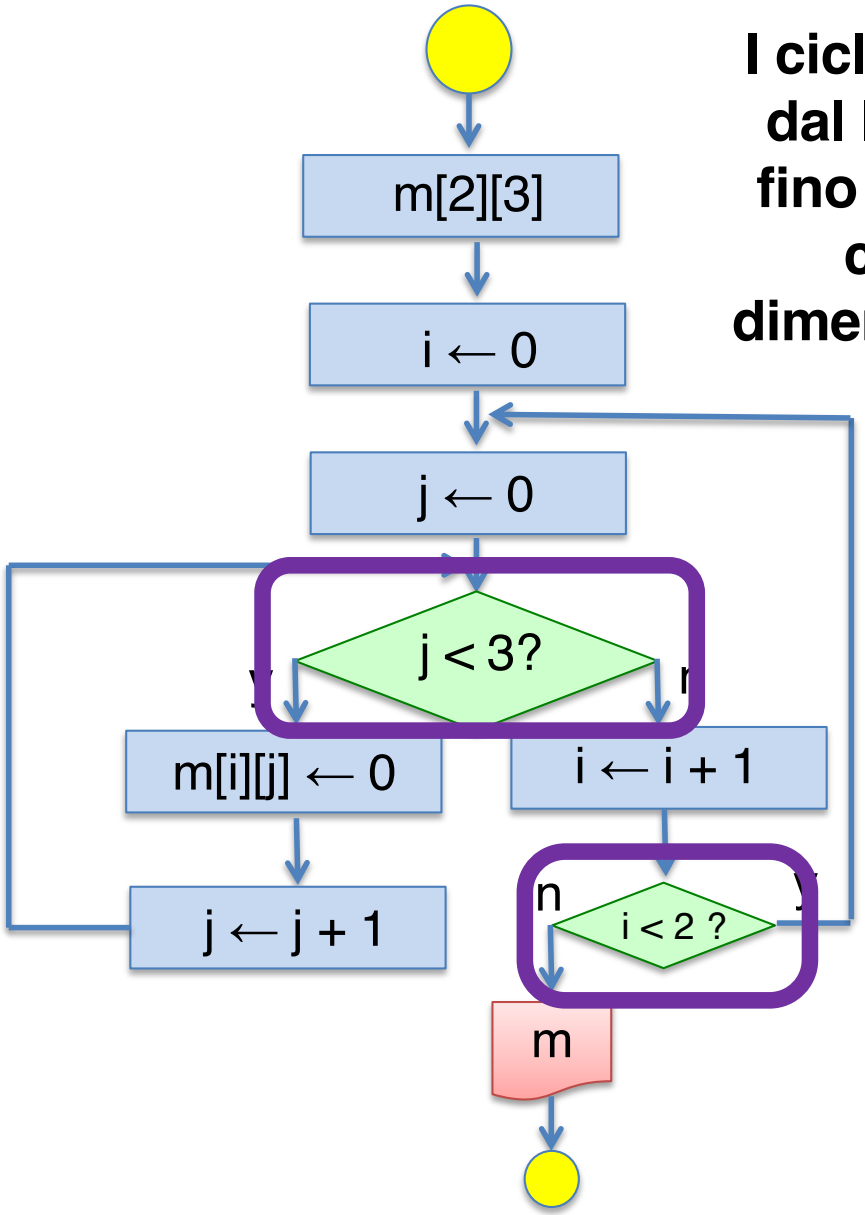
Ciclo “esterno” sull’indice delle righe.
Per ogni valore di i l’indice j delle colonne compie un suo ciclo.

Algoritmo completo

Ciclo “interno” sull’indice delle colonne. Con i fisso, j compie un ciclo.



Algoritmo completo



I cicli degli indici vanno dal loro valore iniziale fino al valore massimo consentito dalle dimensioni della matrice

Variante: la matrice viene riempita colonna per colonna

