



Università degli Studi di Palermo
Dipartimento di Ingegneria Informatica



Informatica di Base - 6 c.f.u.

Anno Accademico 2007/2008

Docente: ing. Salvatore Sorce

Algoritmi

Facoltà di Lettere e Filosofia

Introduzione

- L'informatica è una tra le discipline scientifiche più giovani e stimolanti
- Evoluzione continua:
 - Realtà virtuale
 - Telemedicina
 - Supercalcolatori
 - Monitoraggio ambientale
- Non sempre intuitivamente si riesce a comprendere gli argomenti che sono oggetto dell'informatica

Malintesi

- Primo malinteso
L'informatica è lo studio dei calcolatori
- Secondo malinteso:
L'informatica è lo studio di come scrivere programmi per calcolatori
- Terzo malinteso:
L'informatica è lo studio degli utilizzi e delle applicazioni dei calcolatori e del software
- Fellows and Parberry, Computing Research News, 1993:
L'informatica non riguarda i calcolatori più che l'astronomia i telescopi, la biologia i microscopi o la chimica storte e provette. La scienza non riguarda i dispositivi: riguarda il modo in cui li utilizziamo e che cosa scopriamo utilizzandoli



La definizione di Informatica

- Il concetto centrale nell'informatica è il concetto di *algoritmo*
- Gibbs&Tucker, Communications of the ACM, 1986
Informatica: lo studio degli algoritmi, che comprende:
 - Le loro proprietà formali e matematiche
 - Le loro realizzazioni hardware
 - Le loro realizzazioni linguistiche
 - Le loro applicazioni
- Algoritmo (da Abu Ja'far Muhammad ibn-Musa Al-Khowarizmi, 780-850 a.c.):
 - Una procedura per risolvere matematicamente un problema in un numero finito di passi, che spesso comprende ripetizioni di una operazione. In generale: un metodo passo-passo per eseguire un dato compito.



Rappresentazione di un algoritmo

- Linguaggio naturale
- Pseudocodice
- Diagrammi di flusso
- Linguaggio di programmazione formale



Rappresentazione di un algoritmo

➤ Linguaggio naturale

- Poni il valore della variabile riporto a 0 e della variabile i a 0. Inizia un ciclo dove i indica la colonna di cifre da sommare, partendo dalla prima (per la quale $i=0$). Per ciascun valore di i , calcola $c_i = a_i + b_i + \text{riporto}$. Se il valore ottenuto per c_i è maggiore di dieci allora poni $\text{riporto} = 1$ e sottrai 10 da c_i ; altrimenti poni riporto uguale a 0. Aggiungi 1 ad i , e ripeti. Quando sono finite le cifre da sommare ($i > m-1$) poni la cifra più a sinistra del risultato, c_m , uguale a riporto e stampa il risultato finale, composto dalle cifre $c_m c_{m-1} c_{m-2} \dots c_1 c_0$. Dopo la stampa, l'algoritmo è finito. Fermati.



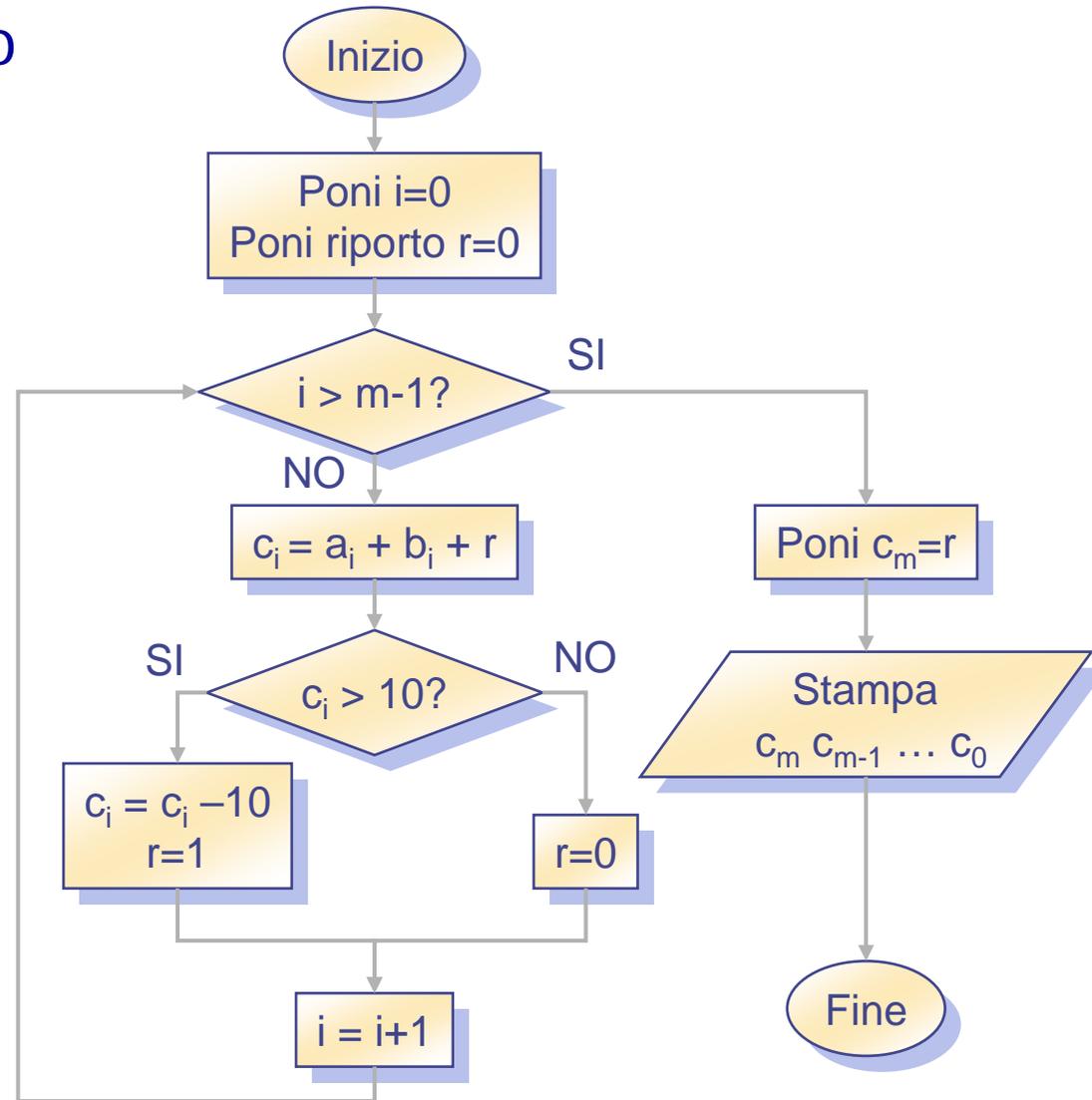
Rappresentazione di un algoritmo

➤ Pseudocodice

1. Poni il valore della variabile riporto a 0
2. Poni il valore della variabile i a 0
3. Ripeti i passi da 4 a 6 fino a che il valore di i è maggiore di $m-1$
4. Somma a_i e b_i al valore corrente del riporto per ottenere c_i
5. Se il valore ottenuto per c_i è maggiore o uguale di 10 allora calcola il nuovo c_i , sottraendo 10 dal valore corrente di c_i
poni riporto=1
altrimenti poni riporto uguale a 0
6. Aggiungi 1 ad i , e ripeti passando alla colonna a sinistra.
7. Poni la cifra più a sinistra del risultato, c_i , uguale a riporto
8. Stampa il risultato finale, $c_m c_{m-1} c_{m-2} \dots c_1 c_0$
9. Fermati.

Rappresentazione di un algoritmo

➤ Diagramma di flusso





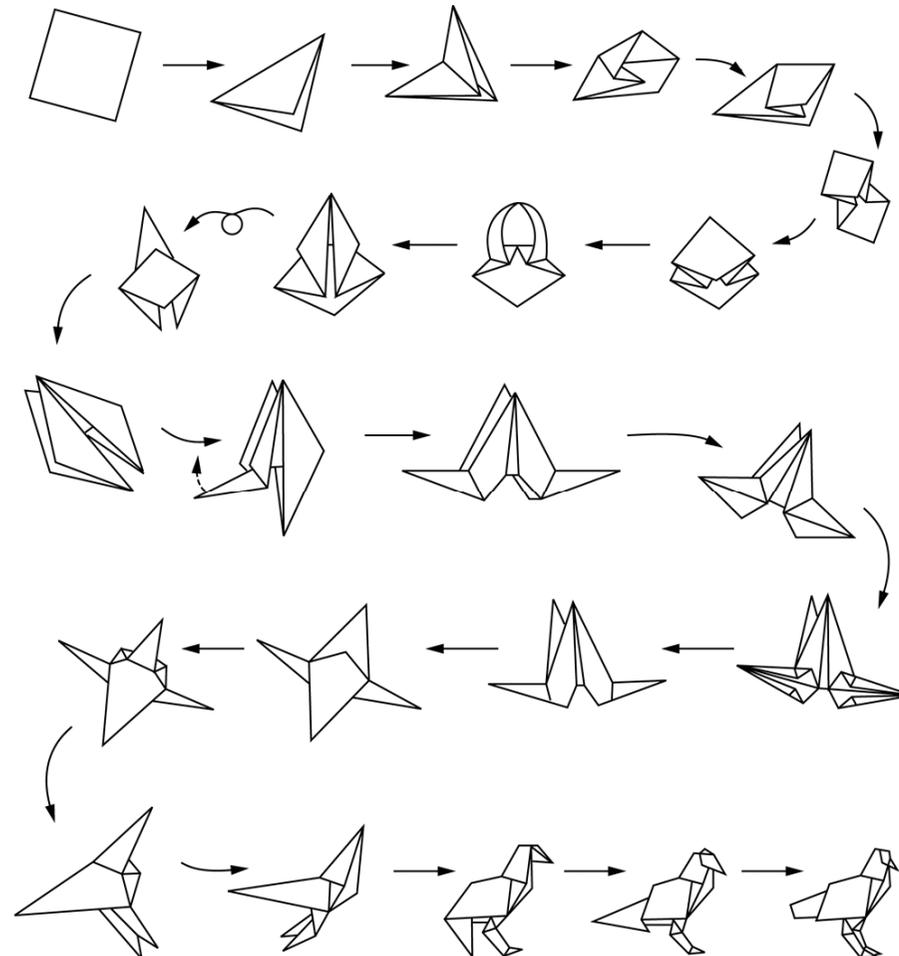
Rappresentazione di un algoritmo

➤ Linguaggio di programmazione formale (C++)

```
1. int a[3], b[3], c[3];
2. int m, i, riporto;
3. riporto=0;
4. i=0;
5. while(i<m) {
6.     c[i]=a[i]+b[i]+riporto;
7.     if(c[i]>10)
8.         {c[i]-=10;
9.         r=1;}
10.    else r=0;
11.    i++;
12. }
13. c[m]=r;
14. printf("%d %d %d\n", c[2],c[1],c[0]);
```



Un esempio di algoritmo – L'origami





Definizione Formale di Algoritmo

- Un insieme ben ordinato di operazioni non ambigue ed effettivamente calcolabili che, eseguito, produce un risultato e termina in una quantità finita di tempo.
 - Insieme ben ordinato
 - Operazioni non ambigue e calcolabili
 - Produce un risultato
 - Termina in una quantità finita di tempo



Insieme ben ordinato

- Ordinamento delle operazioni da eseguire chiaro e non ambiguo
- Il controllo deve procedere senza ambiguità da una operazione alla successiva
 - Le operazioni sono elencate come passi numerati
 - In assenza di altra indicazione (operazioni condizionali o iterative) il controllo passa sempre al passo successivo

Preparazione di una torta di ciliegie

1. Prepara la base
2. Prepara il ripieno di ciliegie
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti



Operazioni non ambigue e calcolabili

- Tutti i passi devono essere chiari per l'agente.
 - I passi 1 e 2 potrebbero essere chiari solo ad un pasticciere professionista.
- Una operazione non-ambigua è detta una operazione ***primitiva***
- Partendo da una prima versione dell'algoritmo, occorre verificare che tutte le operazioni coinvolte siano primitive

Preparazione di una torta di ciliegie

1. Prepara la base
2. Prepara il ripieno di ciliegie
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti



Operazioni non ambigue e calcolabili

- Per un non professionista, il passo 1 va scomposto in un insieme di sottopassi più semplici

➤ Preparazione di una torta di ciliegie

1. Prepara la base

- 1.1. *Prendi 1/3 tazza di farina*
- 1.2. *Setaccia la farina*
- 1.3. *In una terrina, miscela farina, 1/2 tazza di burro e 1/4 tazza di acqua*
- 1.4. *Spiana il composto in due basi di torta da circa 23 cm*

2. Prepara il ripieno di ciliegie

3. Versa il ripieno sulla base

4. Cuoci in forno a 200°C per 45 minuti



Operazioni non ambigue e calcolabili

- Il passo 1 viene scomposto in un insieme di sottopassi più semplici
- Il passo 2 viene scomposto in una serie di sottopassi più semplici
- Tuttavia, il passo 1.2 potrebbe essere non adatto ad essere eseguito per esempio da un bambino

➤ Preparazione di una torta di ciliegie

1. Prepara la base
 - 1.1. Prendi 1/3 tazza di farina
 - 1.2. Setaccia la farina
 - 1.3. In una terrina, miscela farina, 1/2 tazza di burro e 1/4 tazza di acqua
 - 1.4. Spiana il composto in due basi di torta da circa 23 cm
2. **Prepara il ripieno di ciliegie**
 - 2.1. *Versa in una terrina 100 g di ripieno di ciliegia*
 - 2.2. *Aggiungi un pizzico di noce moscata e cannella*
 - 2.3. *Mescola*
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti



Operazioni non ambigue e calcolabili

- Il passo 1.2 può essere ulteriormente decomposto in operazioni più elementari

➤ Preparazione di una torta di ciliegie

1. Prepara la base
 - 1.1. Prendi $\frac{1}{3}$ tazza di farina
 - 1.2. Setaccia la farina**
 - 1.2.1. *Prendi un setaccio e mettilo su una terrina da due quarti di litro*
 - 1.2.2. *Versa la farina nel setaccio e gira la manovella*
 - 1.2.3. *Lascia cadere la farina nella terrina*
 - 1.3. In una terrina, miscela farina, $\frac{1}{2}$ tazza di burro e $\frac{1}{4}$ tazza di acqua
 - 1.4. Spiana il composto in due basi di torta da circa 23 cm
2. Prepara il ripieno di ciliegie
 - 2.1. Versa in una terrina 100 g di ripieno di ciliegia
 - 2.2. Aggiungi un pizzico di noce moscata e cannella
 - 2.3. Mescola
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti



Operazioni non ambigue e calcolabili

- In sintesi, è fondamentale arrivare ad una descrizione dell'algoritmo in primitive eseguibili da un agente di calcolo senza necessità di ulteriori istruzioni.
- Le operazioni devono poi essere effettivamente calcolabili
 - *Stampare la lista di tutti i numeri primi*
 - *Somma 1 al valore corrente di x*

➤ Preparazione di una torta di ciliegie

1. Prepara la base
 - 1.1. Prendi $\frac{1}{3}$ tazza di farina
 - 1.2. Setaccia la farina
 - 1.2.1. Prendi un setaccio e mettilo su una terrina da due quarti di litro
 - 1.2.2. Versa la farina nel setaccio e gira la manovella
 - 1.2.3. Lascia cadere la farina nella terrina
 - 1.3. In una terrina, miscela farina, $\frac{1}{2}$ tazza di burro e $\frac{1}{4}$ tazza di acqua
 - 1.4. Spiana il composto in due basi di torta da circa 23 cm
2. Prepara il ripieno di ciliegie
 - 2.1. Versa in una terrina 100 g di ripieno di ciliegia
 - 2.2. Aggiungi un pizzico di noce moscata e cannella
 - 2.3. Mescola
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti



Produce un risultato

- Gli algoritmi risolvono problemi
- Per comprendere se una soluzione algoritmica è corretta, il suo risultato deve potere essere confrontato con quello atteso.
- Se un risultato non è producibile, l'algoritmo deve produrre un messaggio di errore, attivare un allarme, o fornire una approssimazione del risultato corretto

➤ **Preparazione di una torta di ciliegie**

1. Prepara la base
2. Prepara il ripieno di ciliegie
3. Versa il ripieno sulla base
4. Cuoci in forno a 200°C per 45 minuti

➤ Risultato:

- la torta

➤ **Programmazione di un VCR**

➤ Risultato:

- il nastro col programma televisivo registrato

Termina in una quantità finita di tempo

- Il risultato deve essere prodotto dopo l'esecuzione di un numero finito di operazioni
- E' tipico causare cicli infiniti quando la condizione presente nelle operazioni iterative non si verifica mai
- Esempio:
 1. Poni $i=0$
 2. Ripeti i passi 3 e 4 finché $i < 3$
 3. Stampa "Ciao!"
 4. $i = i + 1$
 5. Fine
- Risultato:
 - Ciao!Ciao!Ciao!
 - oppure
 - Ciao!Ciao!Ciao!Ciao!Ciao!...

- **Ciclo infinito: fare lo shampoo**
 1. Inumidisci i capelli
 2. Insapona
 3. Risciacqua
 4. Ripeti
- Risultato
 - Il cliente si stanca, finisce l'acqua, finisce lo shampoo?
- **Ciclo finito Fare lo shampoo**
 1. Inumidisci i capelli
 2. Ripeti due volte i passi 3 e 4
 3. Insapona
 4. Risciacqua
 5. Stop. Lo shampoo è fatto



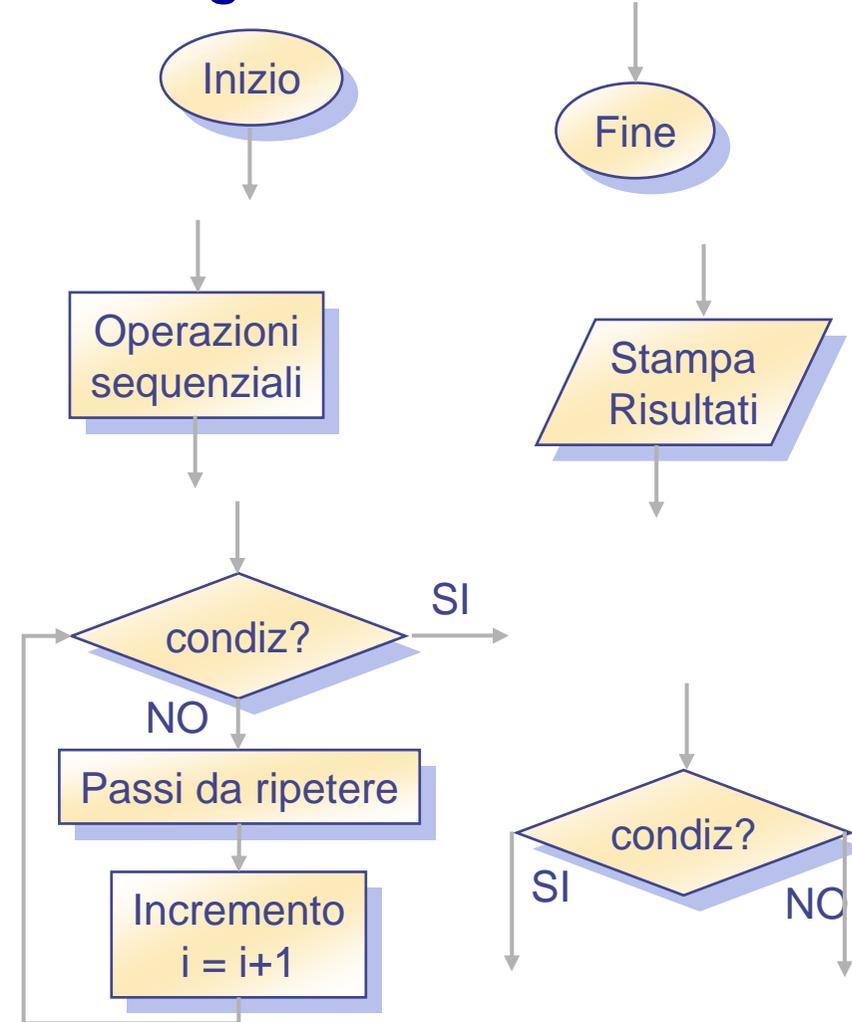
Rappresentazione di un algoritmo

- Linguaggio naturale
- Pseudocodice
- Diagrammi di flusso
- Linguaggio di programmazione formale

Algoritmi e diagrammi di flusso

- **Pseudocodice**
- Operazioni sequenziali
 - Realizza un solo compito ben definito
 - Il controllo passa all'operazione successiva quando il compito è finito
 - Frase dichiarativa
- Operazioni condizionali
 - Selezionano l'operazione successiva sulla base di una domanda
- Operazioni iterative
 - Eseguono un ciclo di istruzioni fino a quando la condizione di uscita è verificata

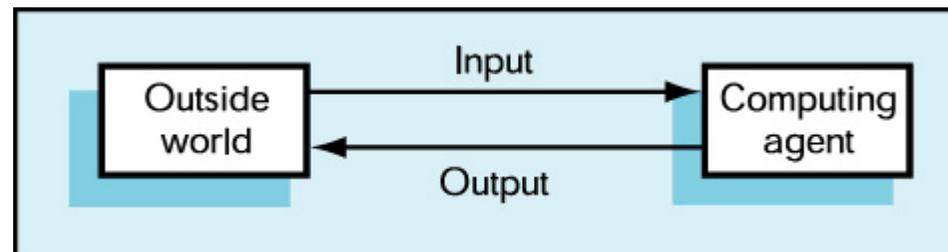
➤ Diagrammi di flusso





Operazioni di ingresso e uscita

- Operazioni sequenziali
 - Elaborazione, Ingresso, Uscita
- Operazioni di ingresso
 - Acquisisci il valore per “variabile”
- Operazioni di uscita
 - Stampa il valore per “variabile”, descrizione della variabile



Somma di due numeri

➤ Pseudocodice

1. Poni il valore della variabile riporto a 0
2. Poni il valore della variabile i a 0
3. Ripeti i passi da 4 a 6 fino a che il valore di i è maggiore di $m-1$
4. Somma a_i e b_i al valore corrente del riporto per ottenere c_i
5. Se il valore ottenuto per c_i è maggiore o uguale di 10 allora calcola il nuovo c_i , sottraendo 10 dal valore corrente di c_i
poni riporto=1
altrimenti poni riporto uguale a 0
6. Aggiungi 1 ad i , e ripeti passando alla colonna a sinistra.
7. Poni la cifra più a sinistra del risultato, c_i , uguale a riporto
8. Stampa il risultato finale, $c_m c_{m-1} c_{m-2} \dots c_1 c_0$
9. Fermati.

Esempio

m=2		
A=47	a1=4	a0=7
B=25	b1=2	b0=5
C=72	c1=7	c0=2
i	Operazioni	Risultato
0	r=0	r=0
	i>m-1?	NO
	c0=a0+b0+r	c0=7+5+0=12
	c0>10?	SI
	c0=c0-10	c0=2
	r=1	r=1

m=2		
A=47	a1=4	a0=7
B=25	b1=2	b0=5
C=72	c1=7	c0=2
i	Operazioni	Risultato
1	r=1	r=1
	i>m-1?	NO
	c1=a1+b1+r	c1=4+2+1=7
	c1>10?	NO
	r=0	r=0

Esempio

m=2		
A=47	a1=4	a0=7
B=25	b1=2	b0=5
C=72	c1=7	c0=2
i	Operazioni	Risultato
2	i>m-1?	SI
	c2=r	c2=0
	C=c2 c1 c0	C=072



Rappresentazione di un algoritmo

- Linguaggio naturale
- Pseudocodice
- Diagrammi di flusso
- Linguaggio di programmazione formale

Somma di Due Numeri

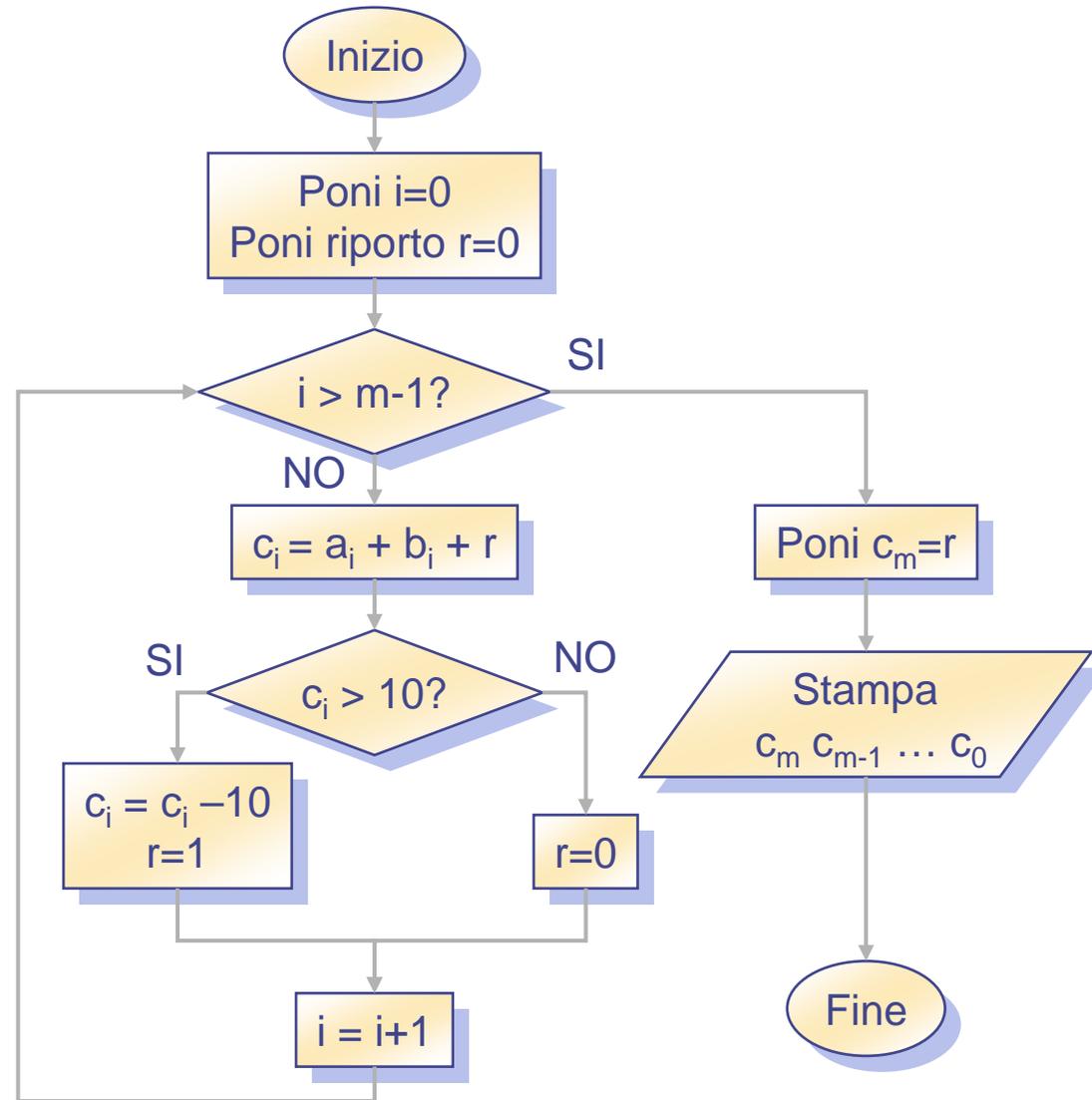
- La somma di due numeri:

$$\begin{array}{r} 1 \\ 47 + \\ 25 = \\ \hline 72 \end{array}$$

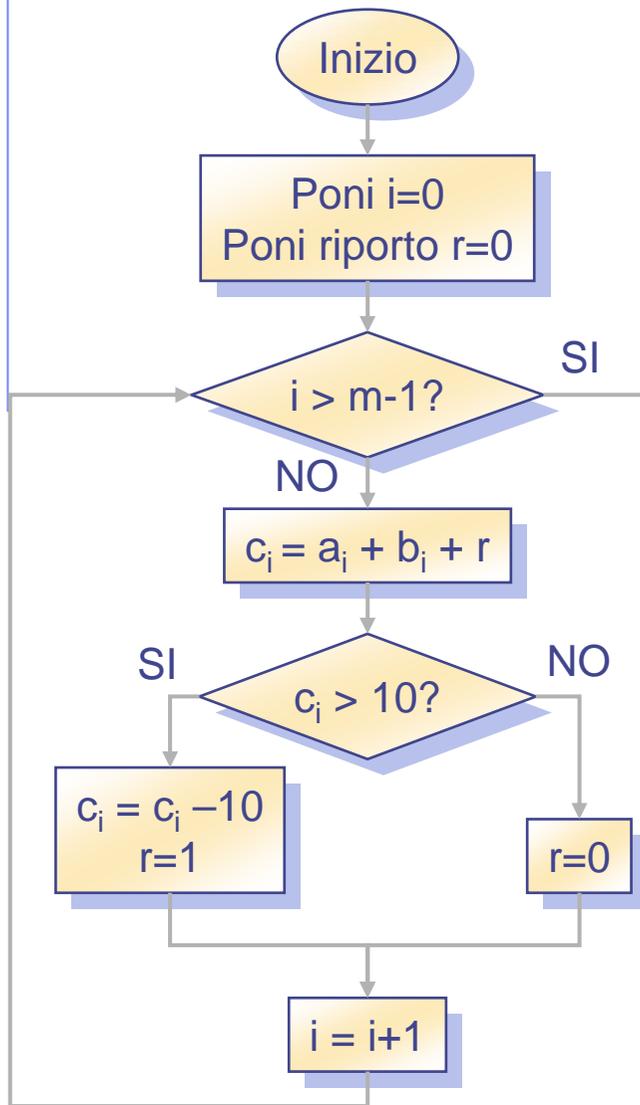
- Due numeri con $m > 1$ cifre:

$$A = a_{m-1} a_{m-2} \dots a_0 +$$

$$B = b_{m-1} b_{m-2} \dots b_0 =$$

$$C = c_m c_{m-1} c_{m-2} \dots c_0$$


Somma di Due Numeri (modificato)

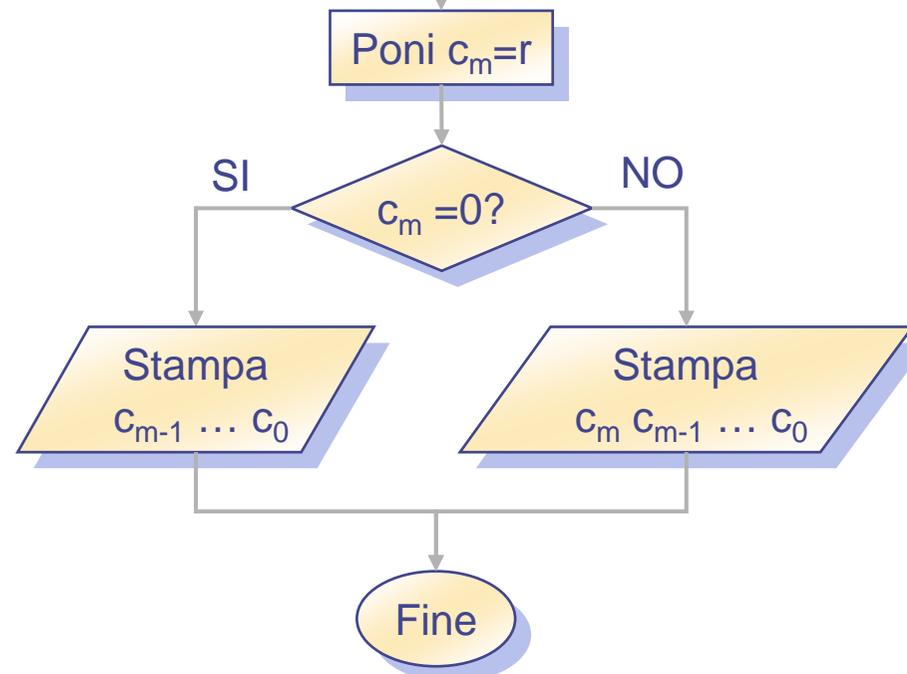


➤ Se $c_m = 0$ allora

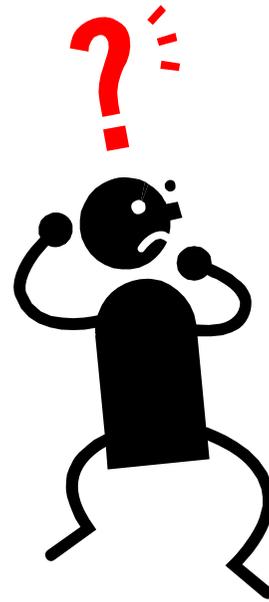
$$C = c_{m-1} c_{m-2} \dots c_0$$

altrimenti

$$C = c_m c_{m-1} c_{m-2} \dots c_0$$



Domande?





Esercizio

- Calcolo del Massimo Comun Divisore (algoritmo di Euclide)
 1. Prendi due numeri interi positivi I e J, con $I > J$
 2. Se J è diverso da zero, allora esegui i passi da 3 a 5
 3. Dividi I per J e chiama R il resto
 4. Se R è diverso da 0, reimposta I al valore di J, reimposta J al valore di R e torna al passo 2
 5. Stampa il risultato, MCD che è il valore di J
 6. Stop

$I = 32, J = 20$

$I = 32, J = 0$