

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Informatica di base 2025/2026

WRAP-UP

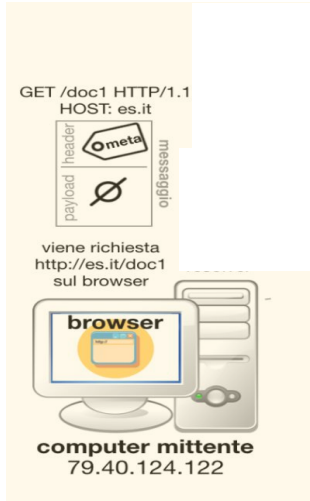
Ivan Heibi

Dipartimento di Filologia Classica e Italianistica (FICLIT)

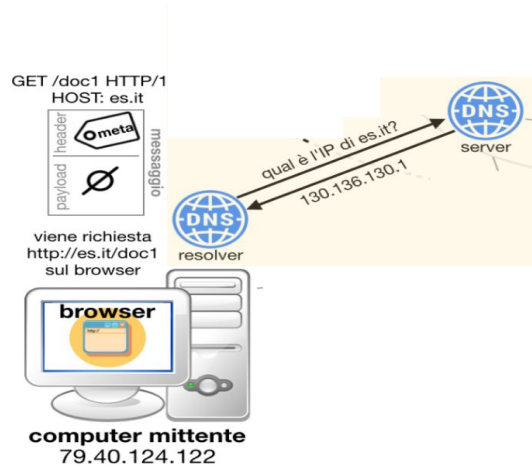
Ivan.heibi2@unibo.it

<https://www.unibo.it/sitoweb/ivan.heibi2>

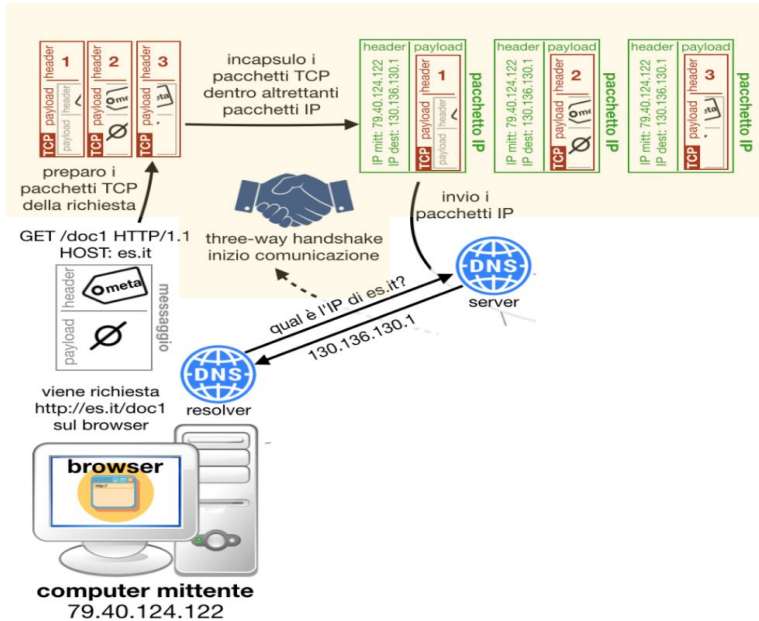
Comunicazione client - web server



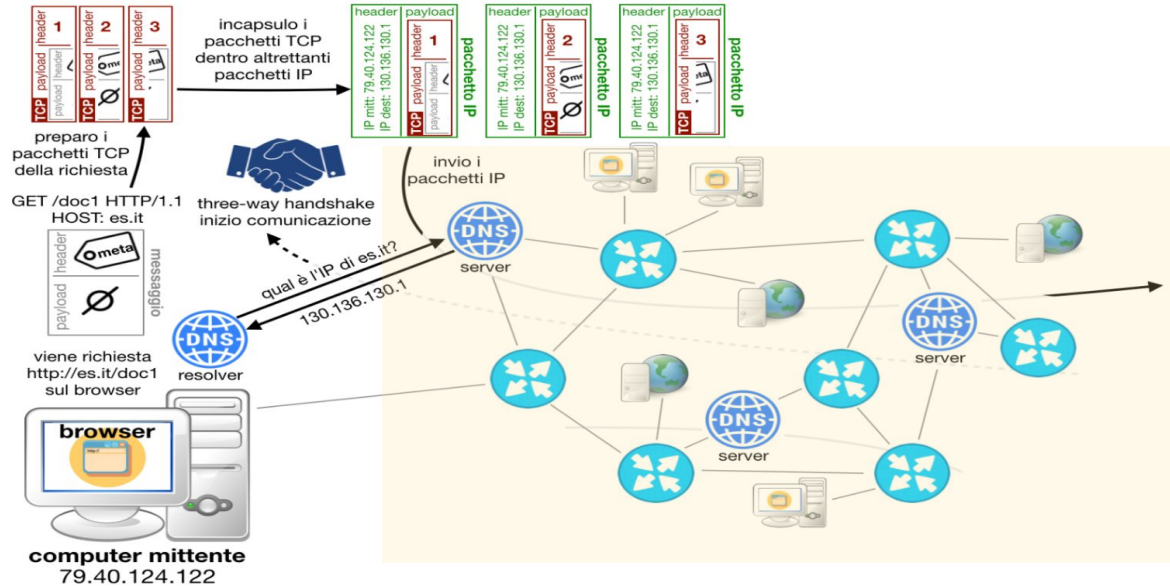
Comunicazione client - web server



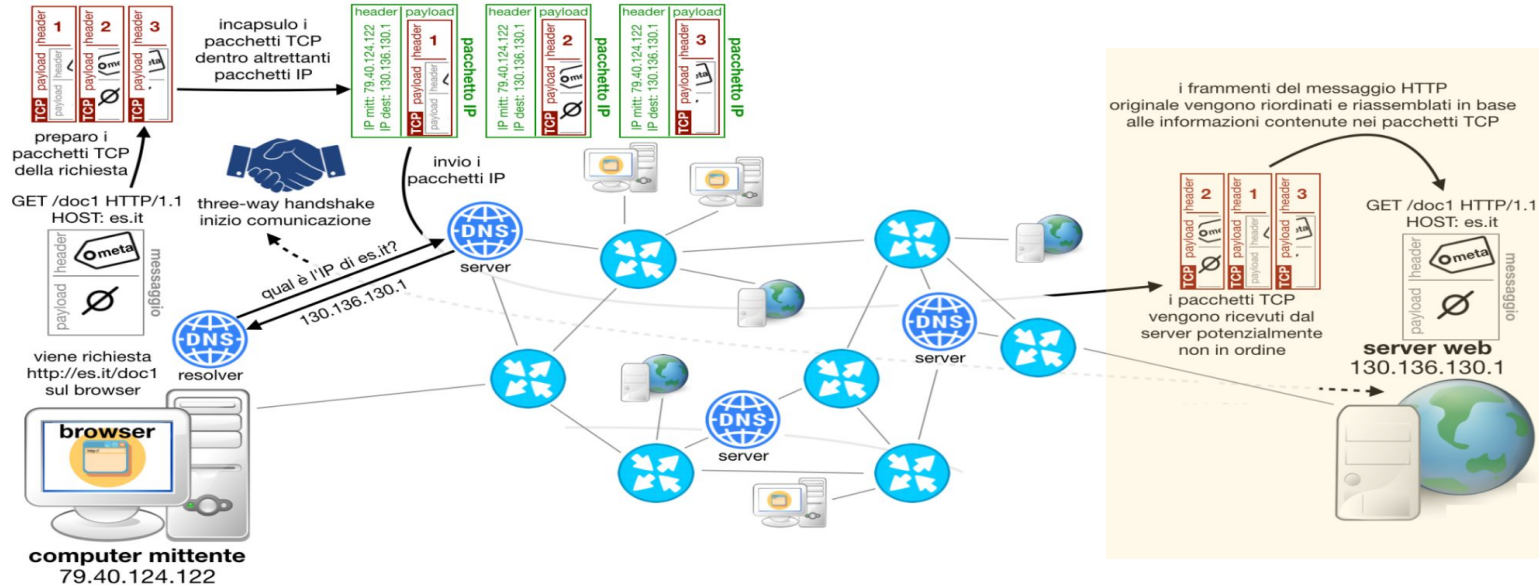
Comunicazione client - web server



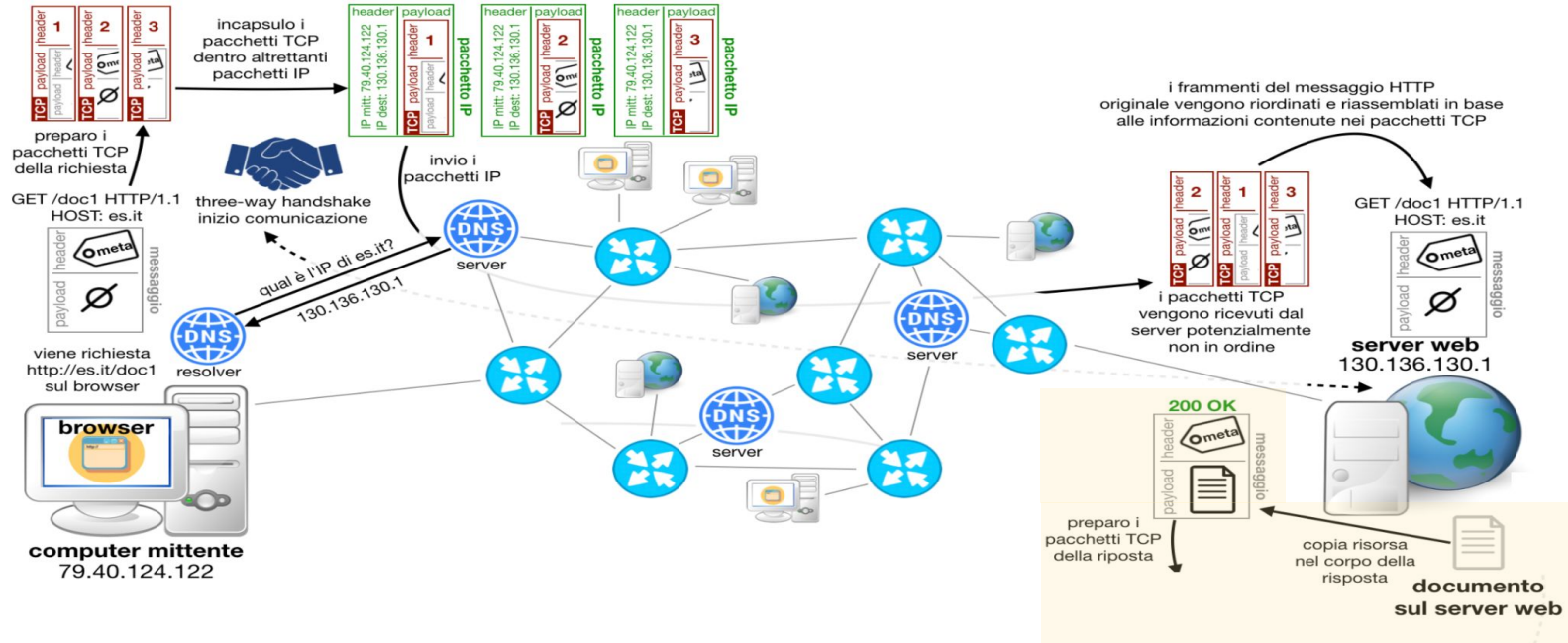
Comunicazione client - web server



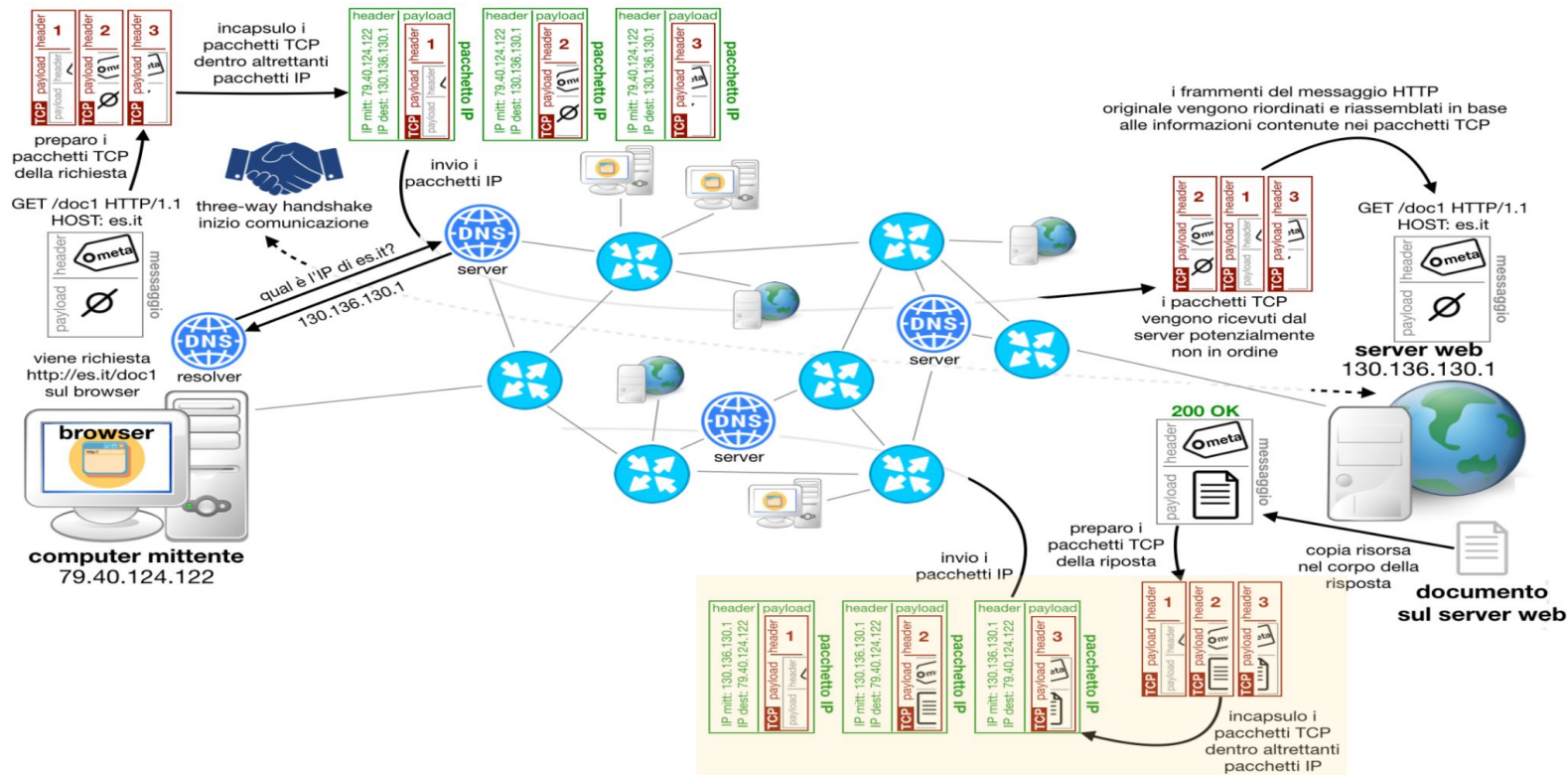
Comunicazione client - web server



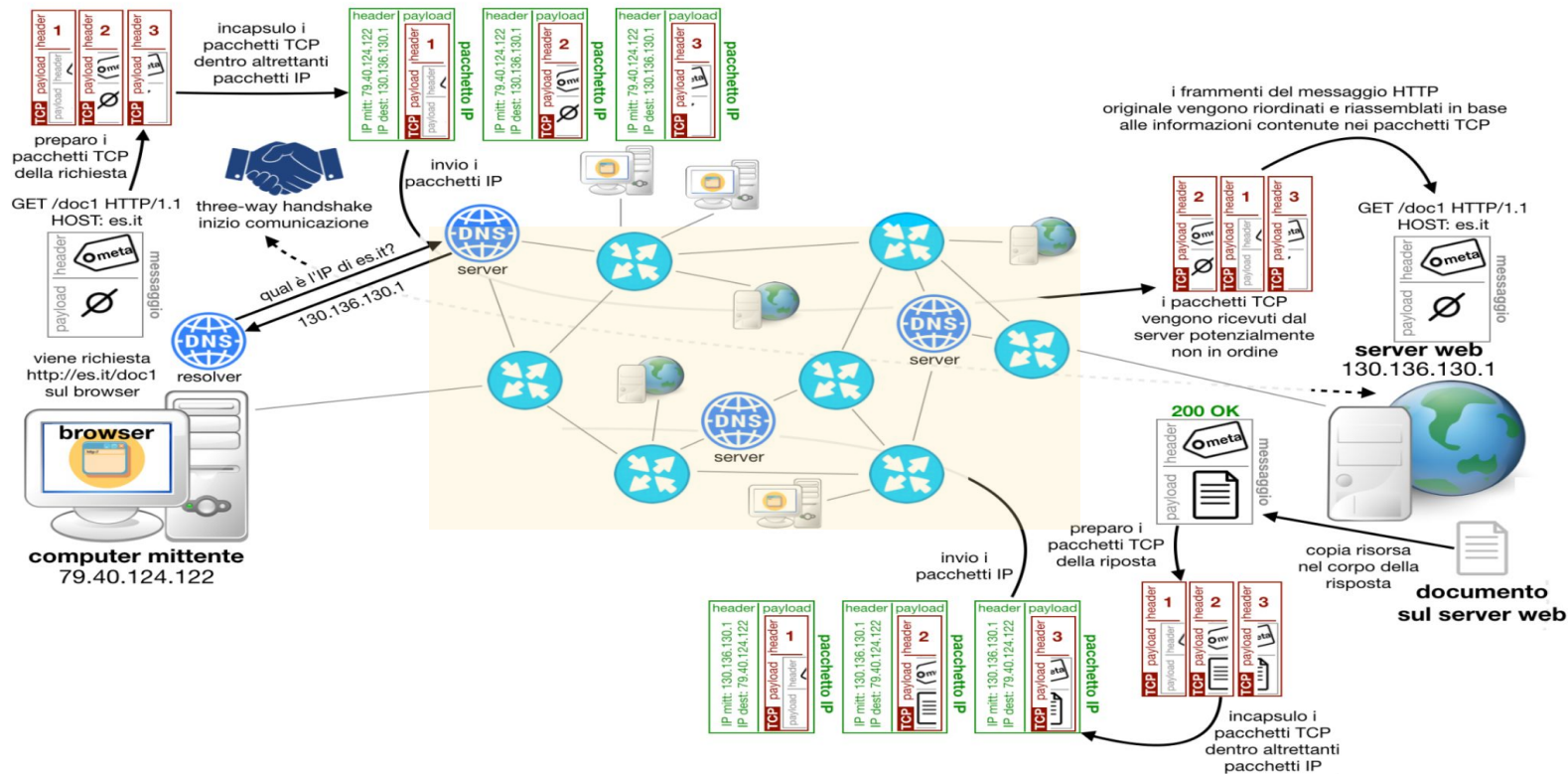
Comunicazione client - web server



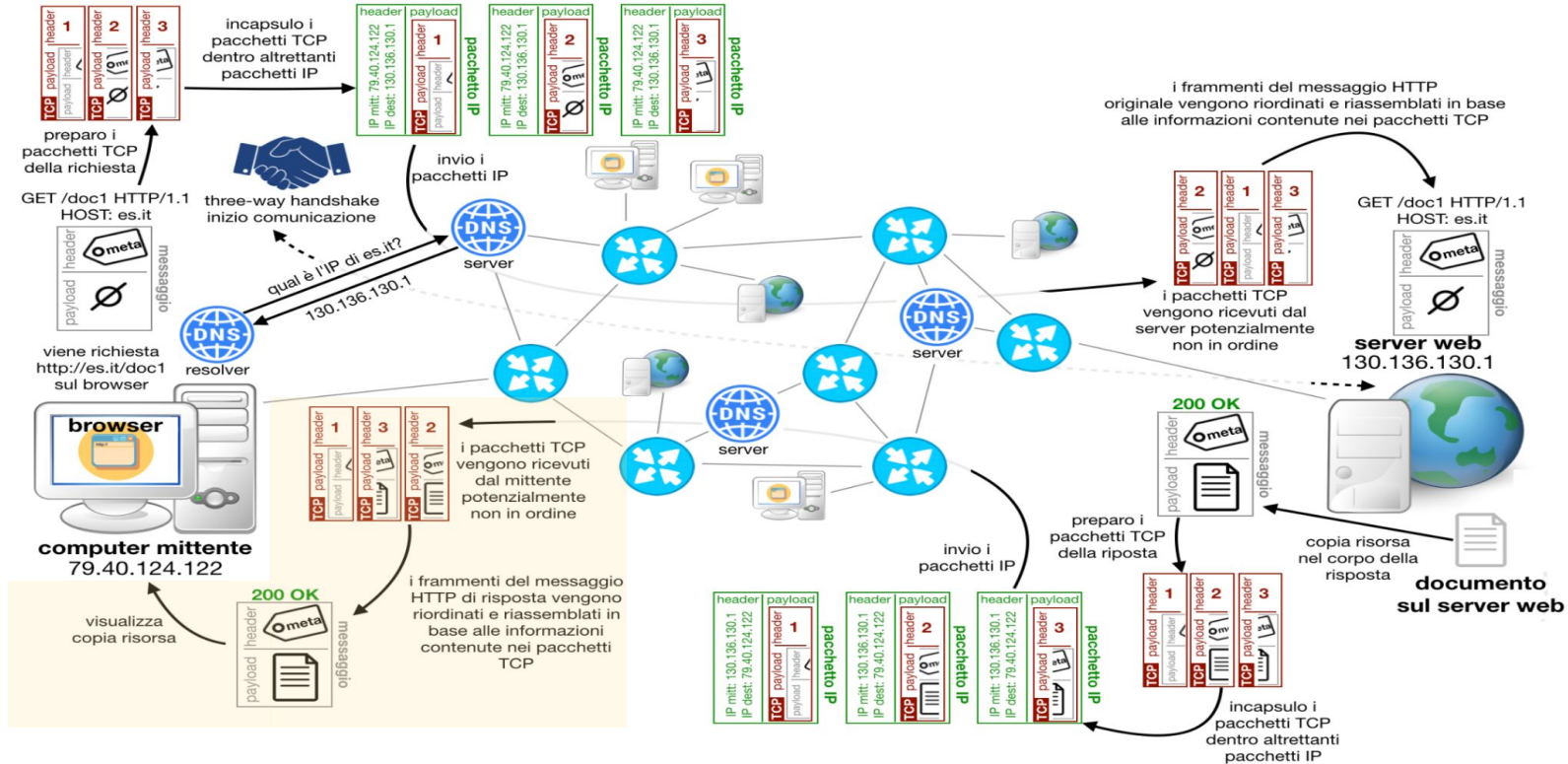
Comunicazione client - web server



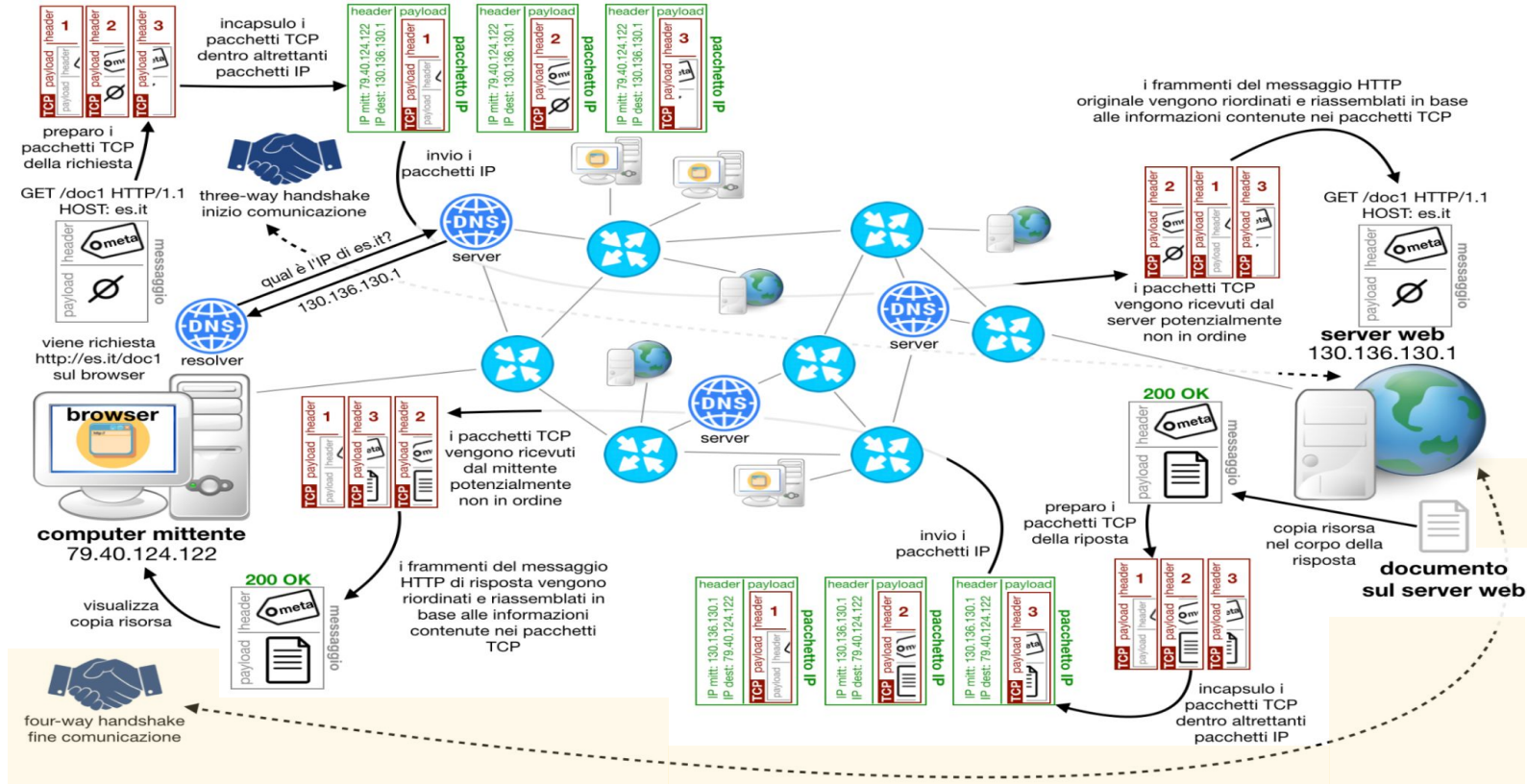
Comunicazione client - web server



Comunicazione client - web server



Comunicazione client - web server



Comunicazione client - web server

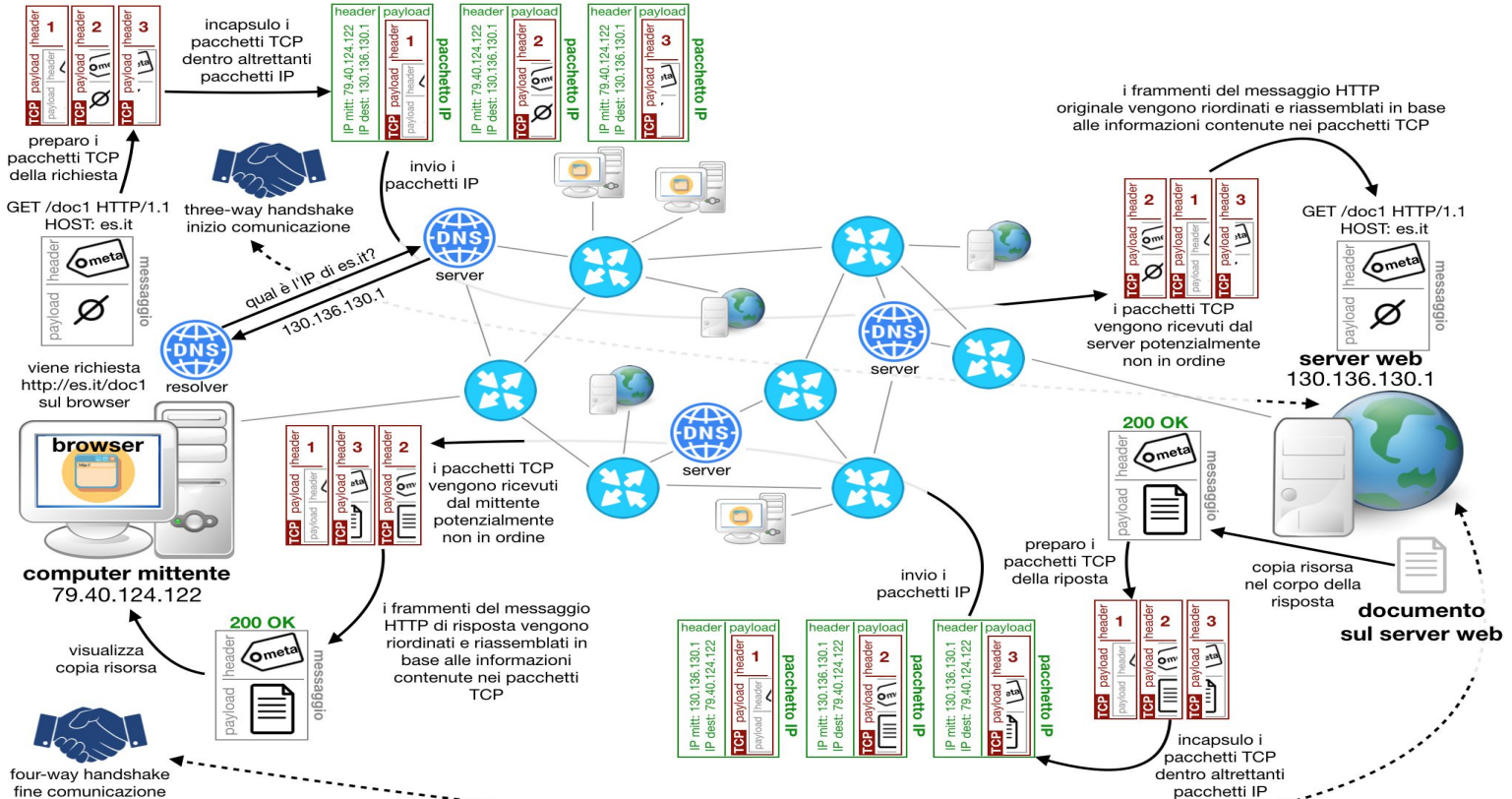


Diagramma di flusso – esercizio

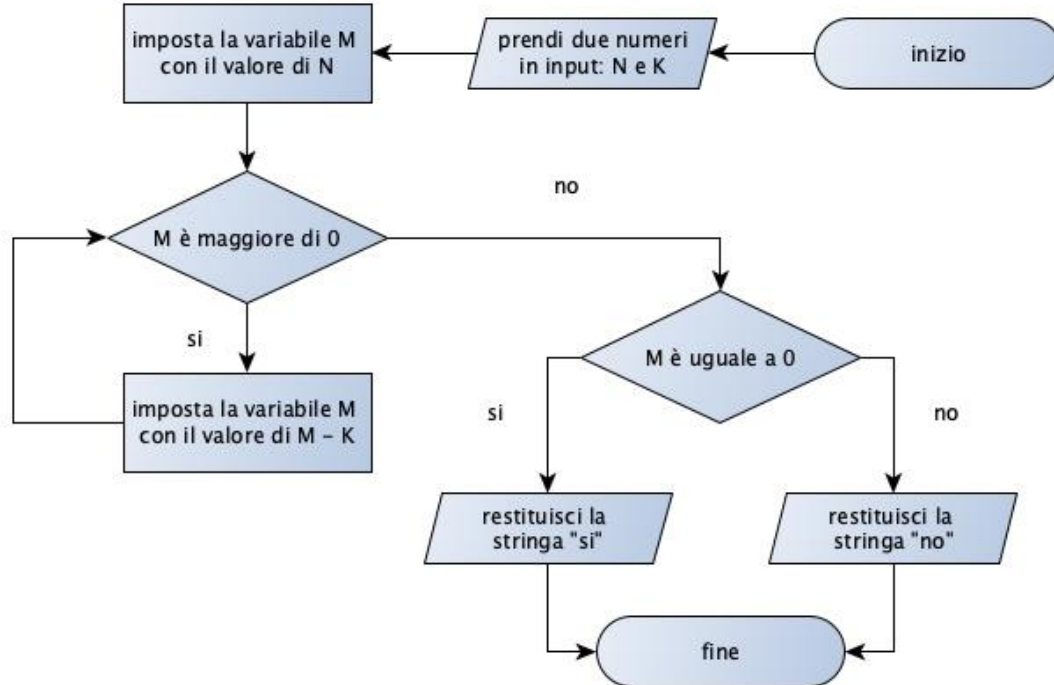


Diagramma di flusso – esercizio

N	15
K	4

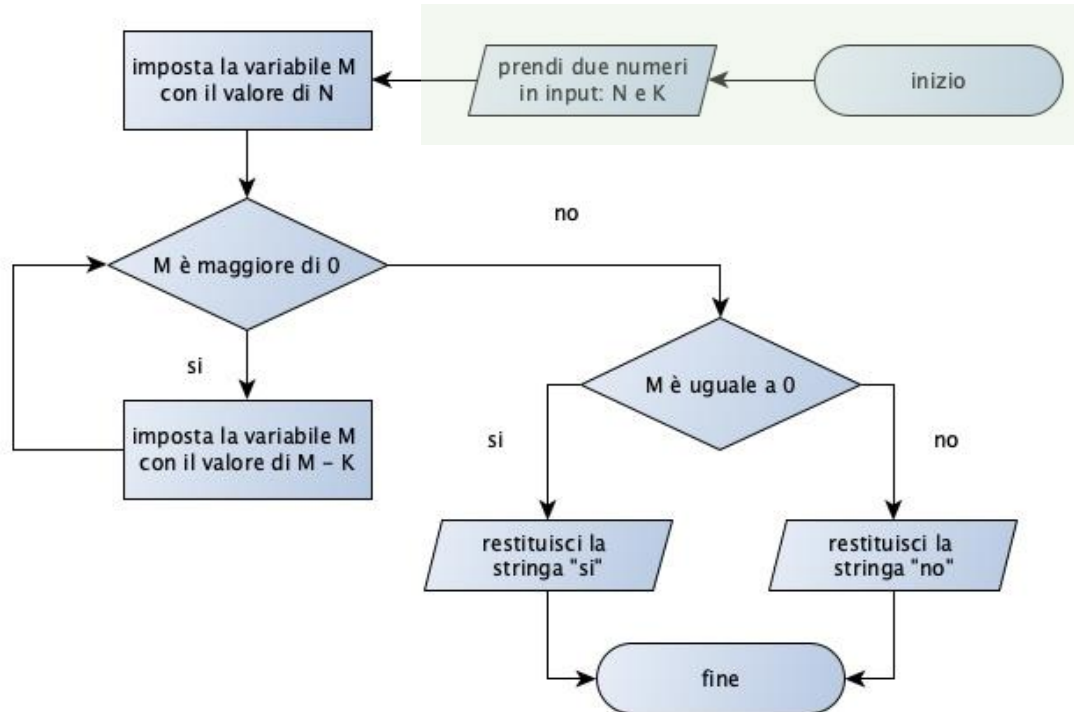


Diagramma di flusso – esercizio

N	15
K	4
M	15

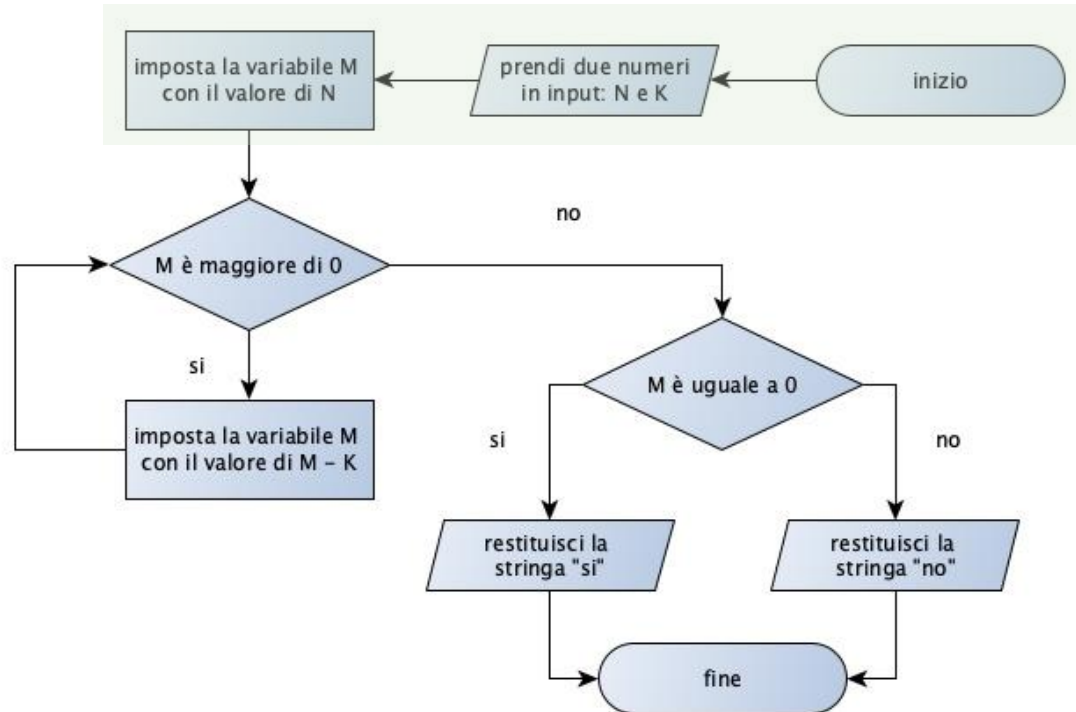


Diagramma di flusso – esercizio

N	15
K	4
M	11

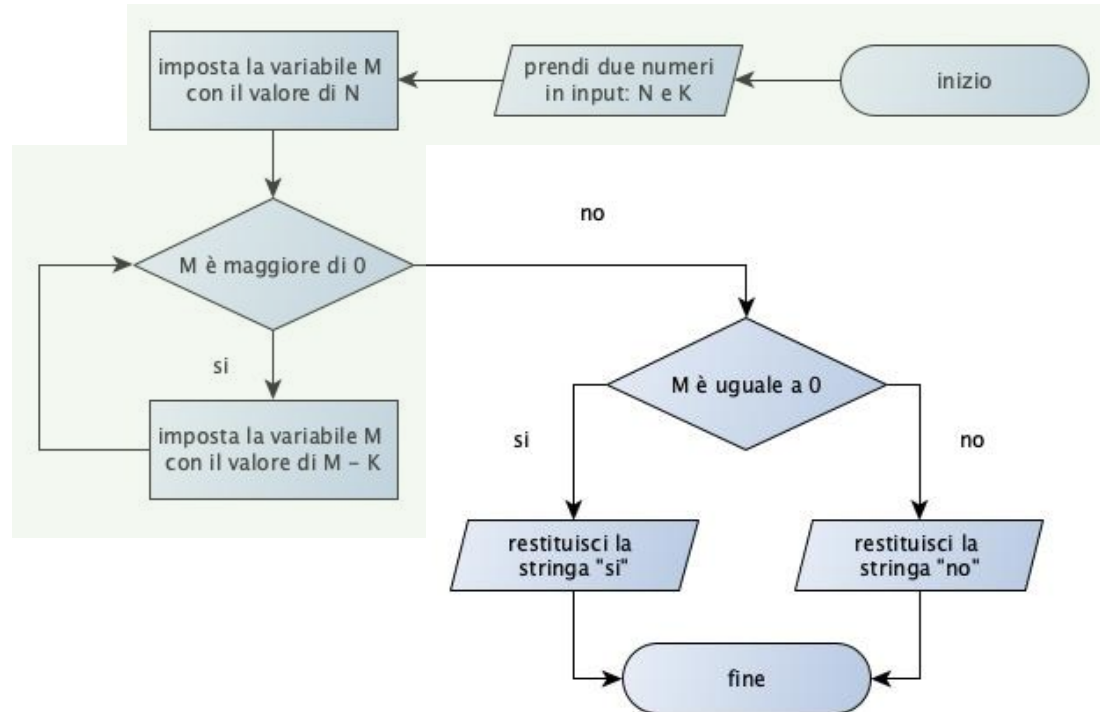


Diagramma di flusso – esercizio

N	15
K	4
M	7

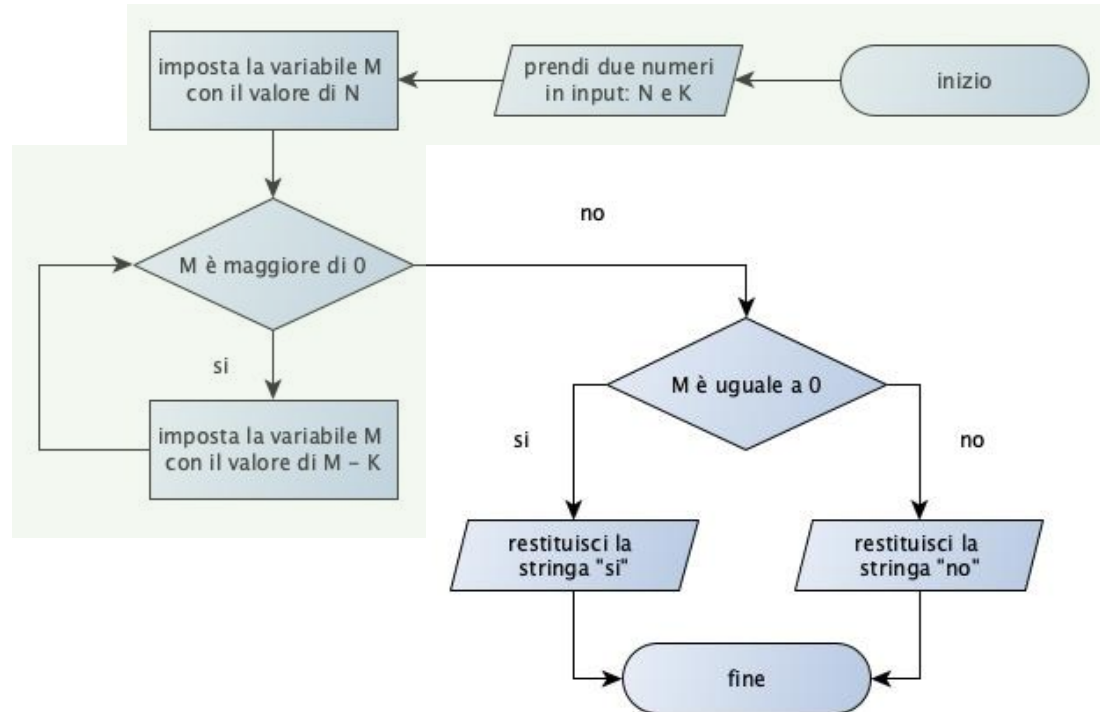


Diagramma di flusso – esercizio

N	15
K	4
M	3

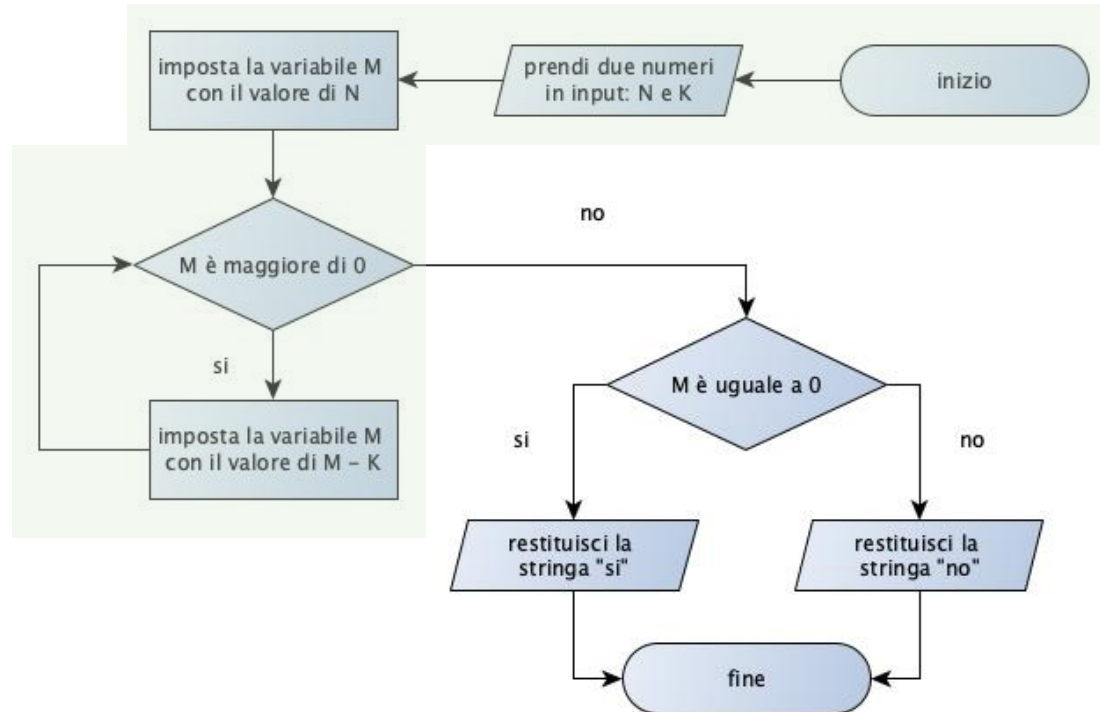


Diagramma di flusso – esercizio

N	15
K	4
M	-1

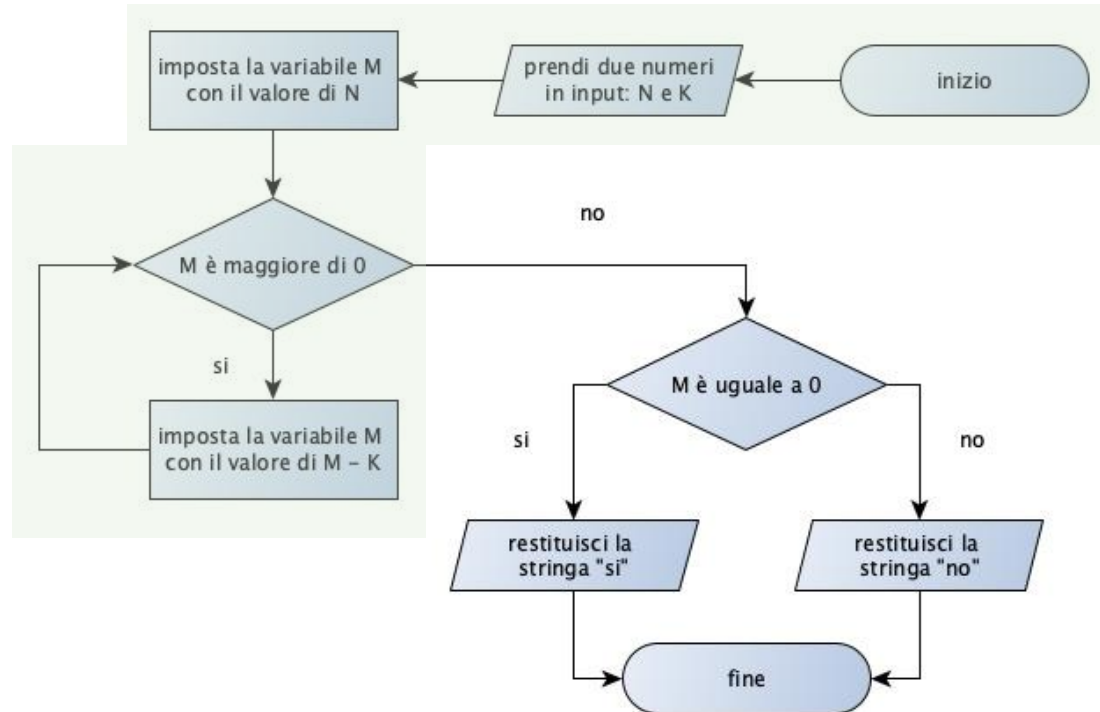


Diagramma di flusso – esercizio

N	15
K	4
M	-1

no

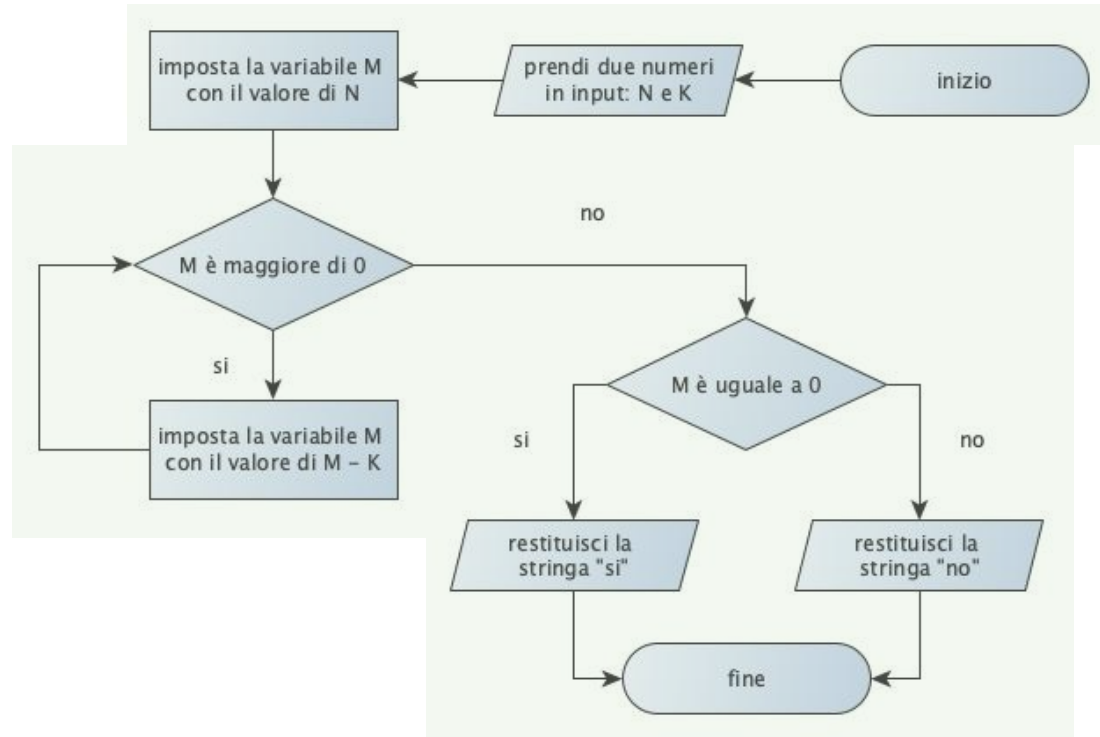
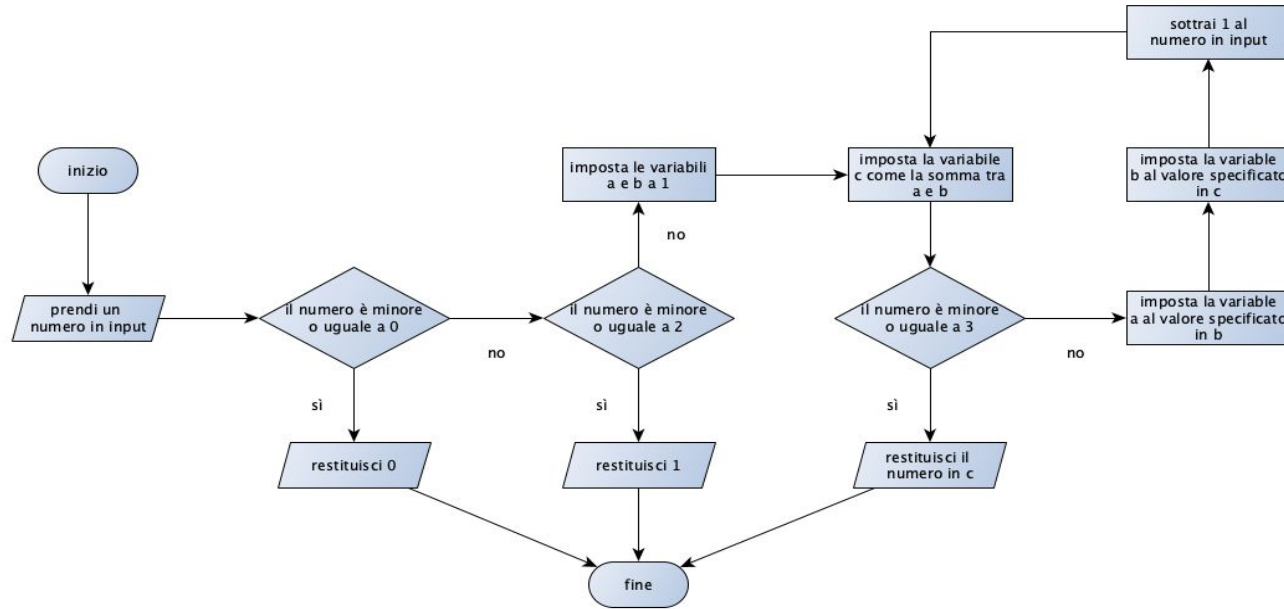
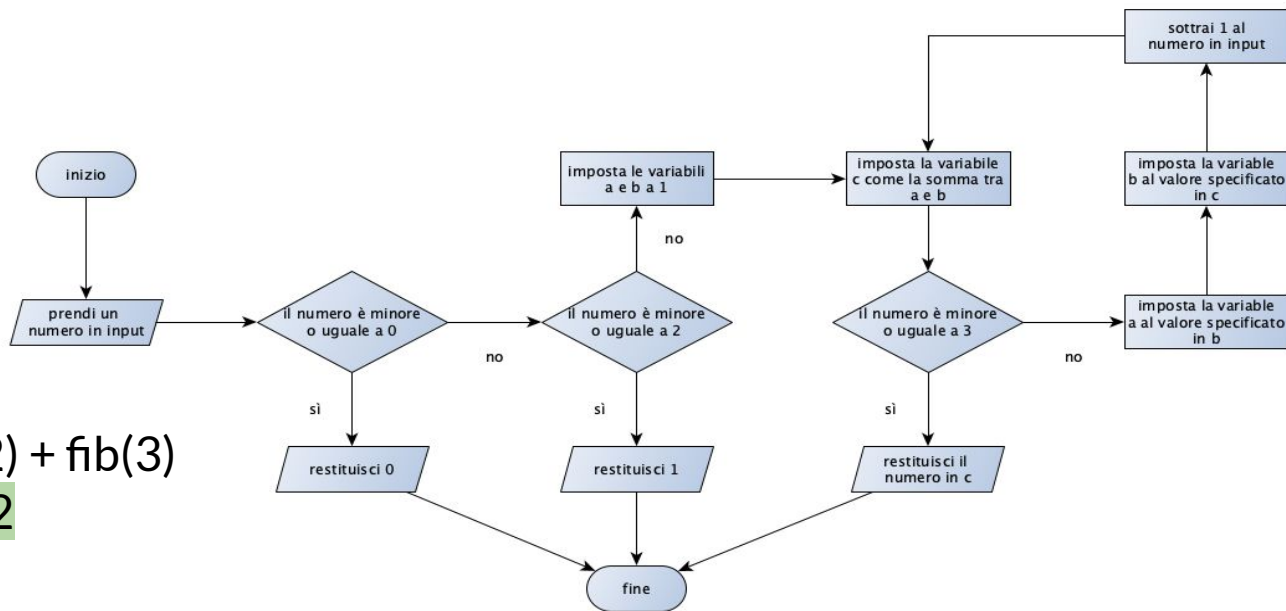


Diagramma di flusso di fib(n)



$n =$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
fib(n) =	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...

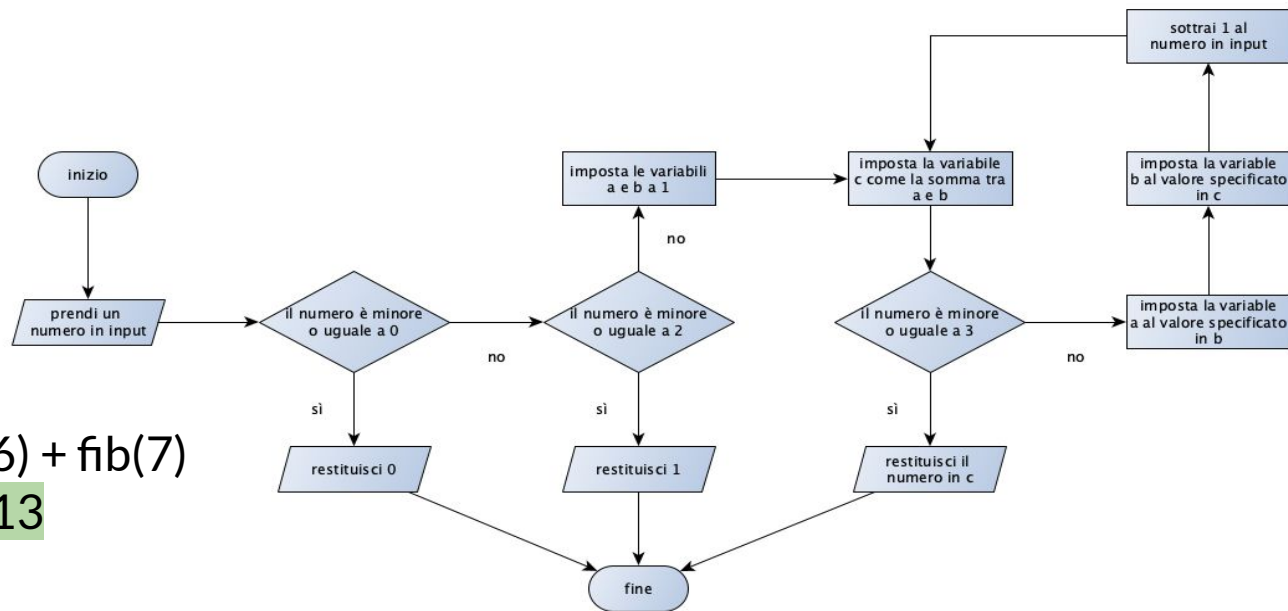
Diagramma di flusso di fib(n)



$$\begin{aligned}
 \text{fib}(4) &= \text{fib}(2) + \text{fib}(3) \\
 &= 1 + 2 \\
 &= 3
 \end{aligned}$$

<i>n</i> =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
fib(<i>n</i>) =	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...

Diagramma di flusso di fib(n)



$$\begin{aligned}
 \text{fib}(8) &= \text{fib}(6) + \text{fib}(7) \\
 &= 8 + 13 \\
 &= 21
 \end{aligned}$$

<i>n</i> =	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	...
fib(n) =	0	1	1	2	3	5	8	13	21	34	55	89	144	233	377	...

Diagramma di flusso fib(n) per n = 6

numero = 6

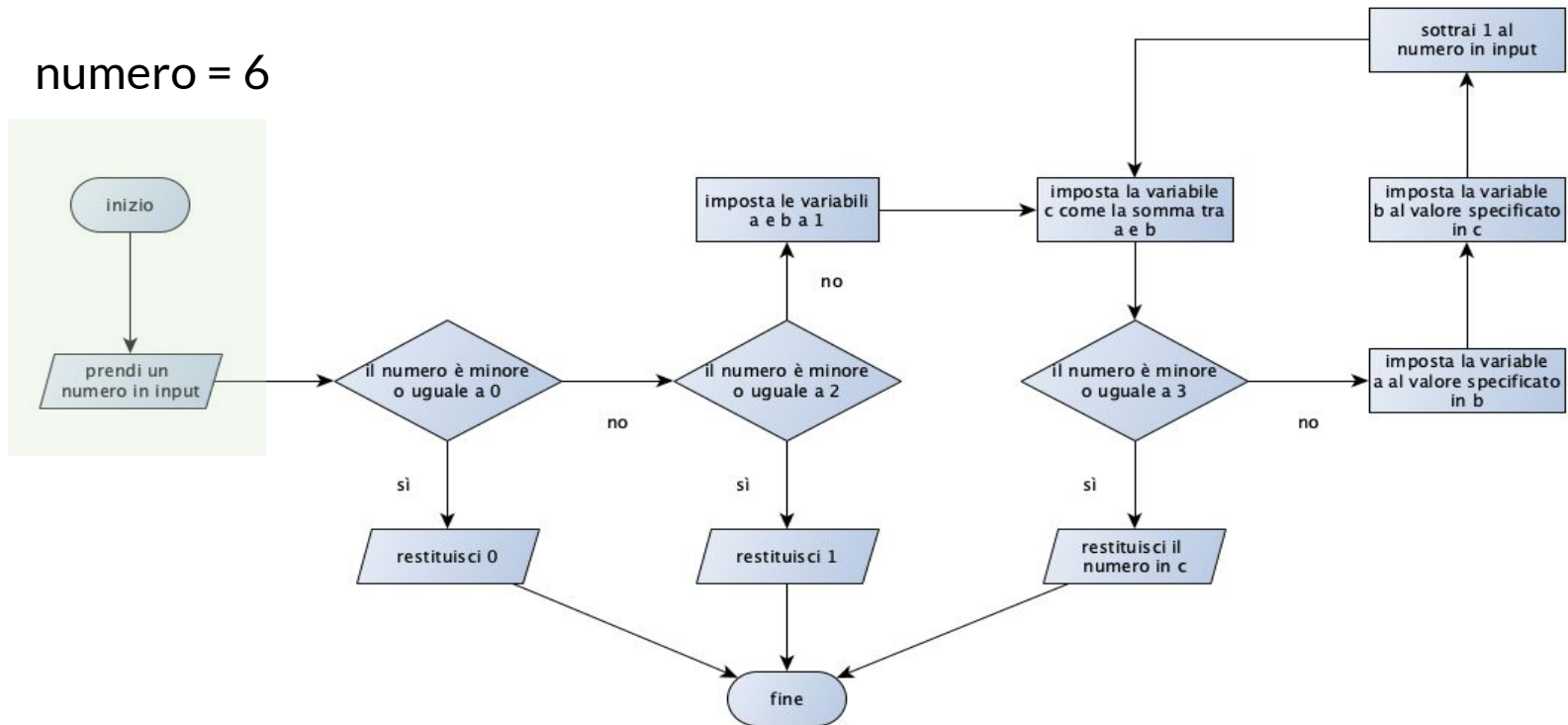


Diagramma di flusso fib(n) per n = 6

numero = 6

a	1
b	1

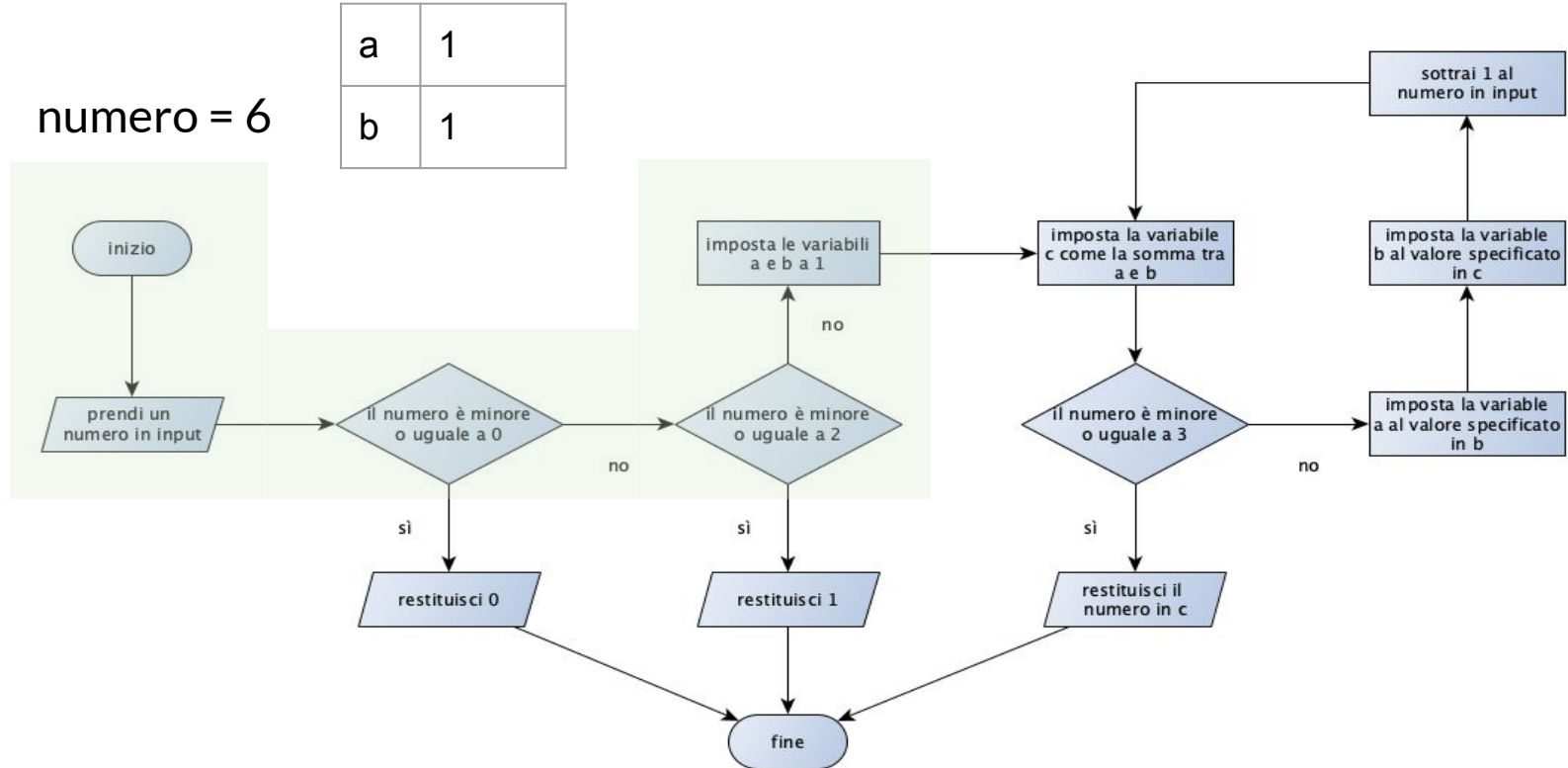


Diagramma di flusso fib(n) per n = 6

numero = 6

a	1
b	1
c	2

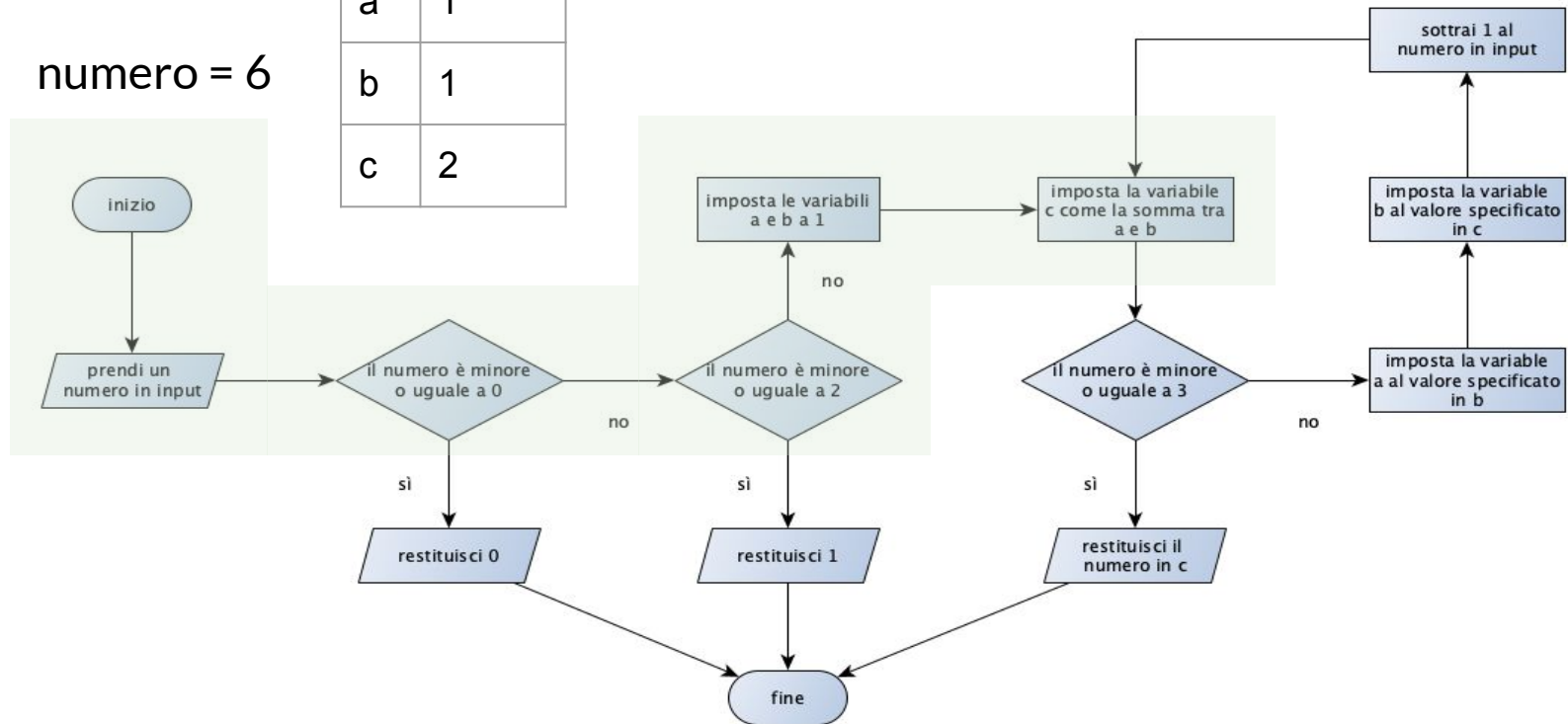


Diagramma di flusso fib(n) per n = 6

numero = 6

a	1
b	1
c	2

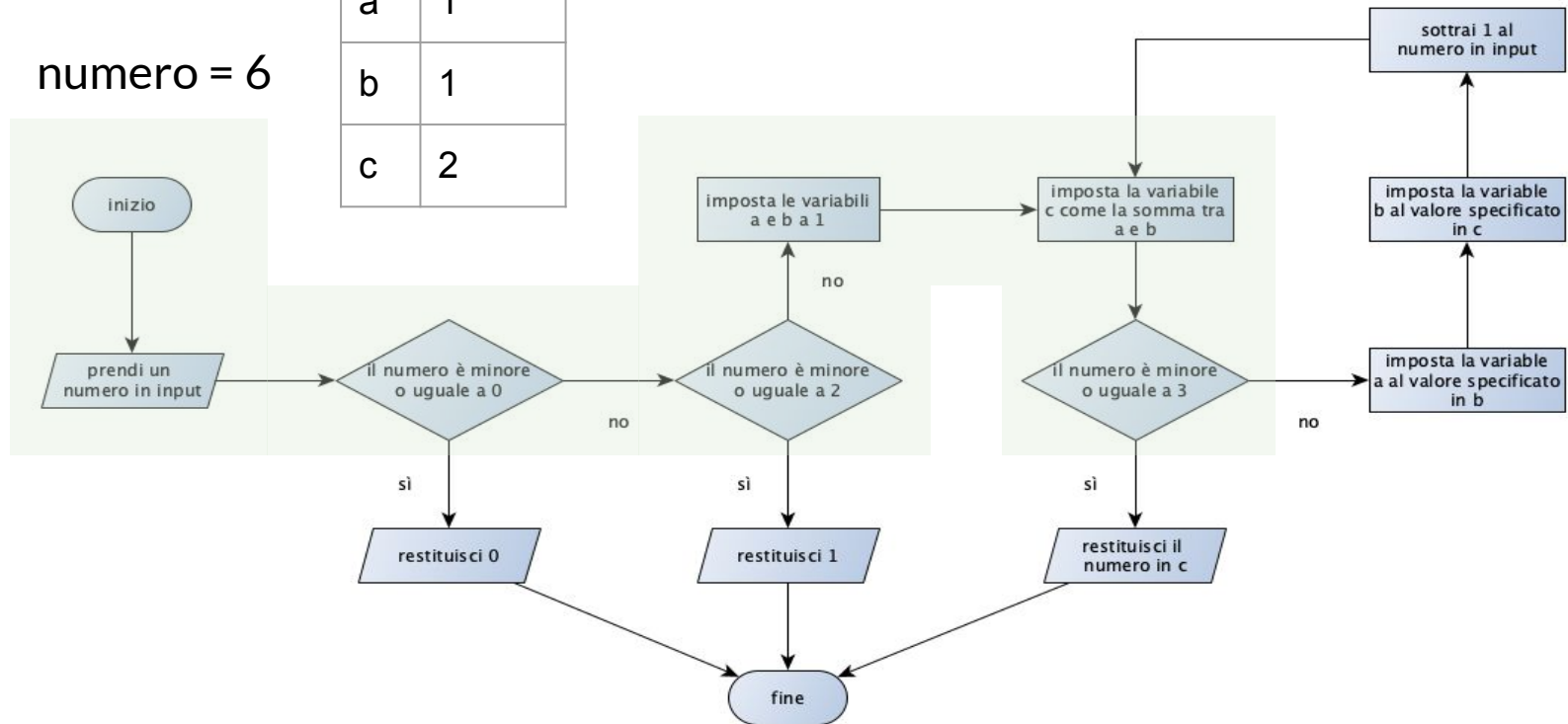


Diagramma di flusso fib(n) per n = 6

numero = 5

a	1
b	2
c	2

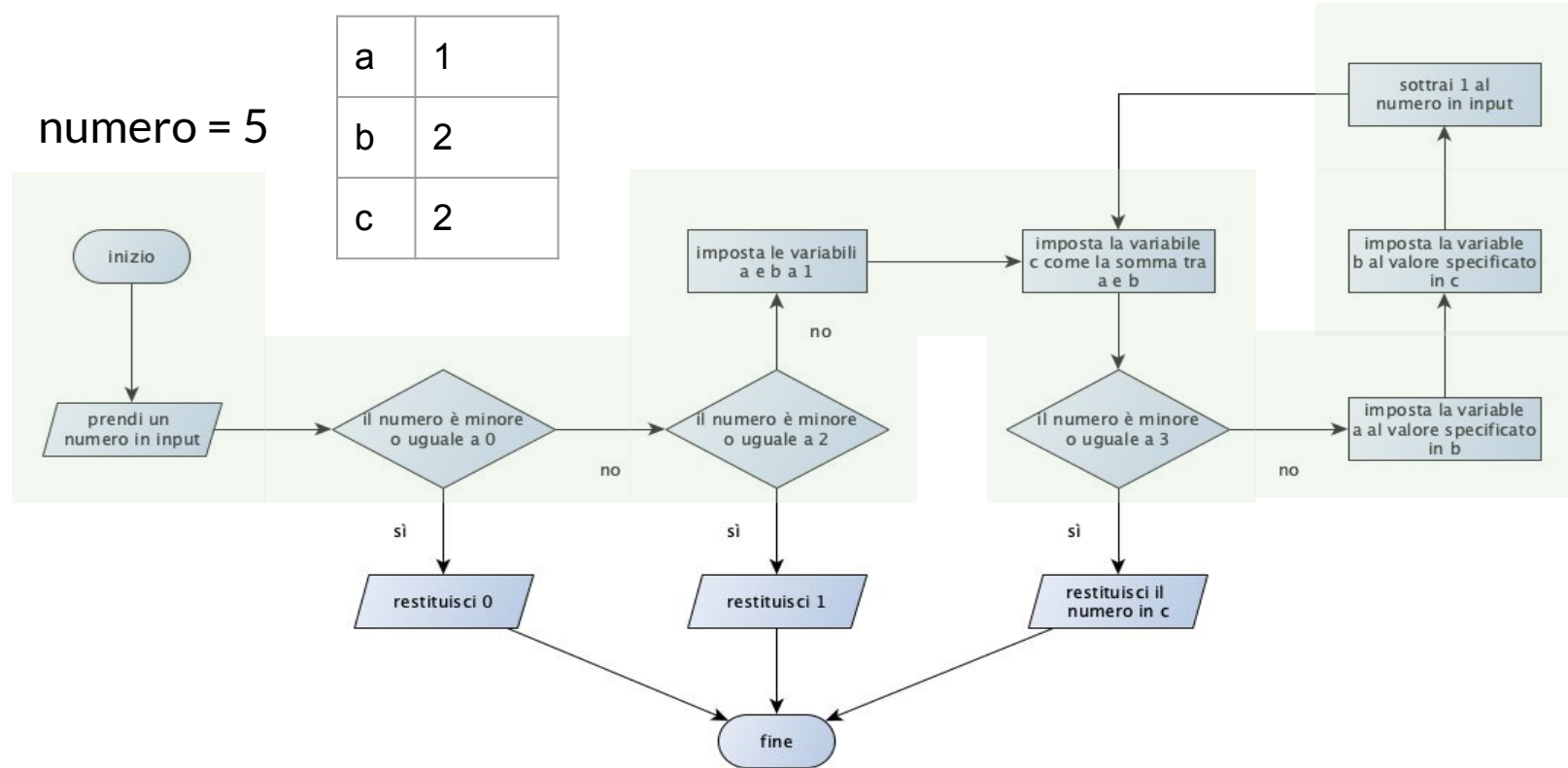


Diagramma di flusso fib(n) per n = 6

numero = 5

a	1
b	2
c	3

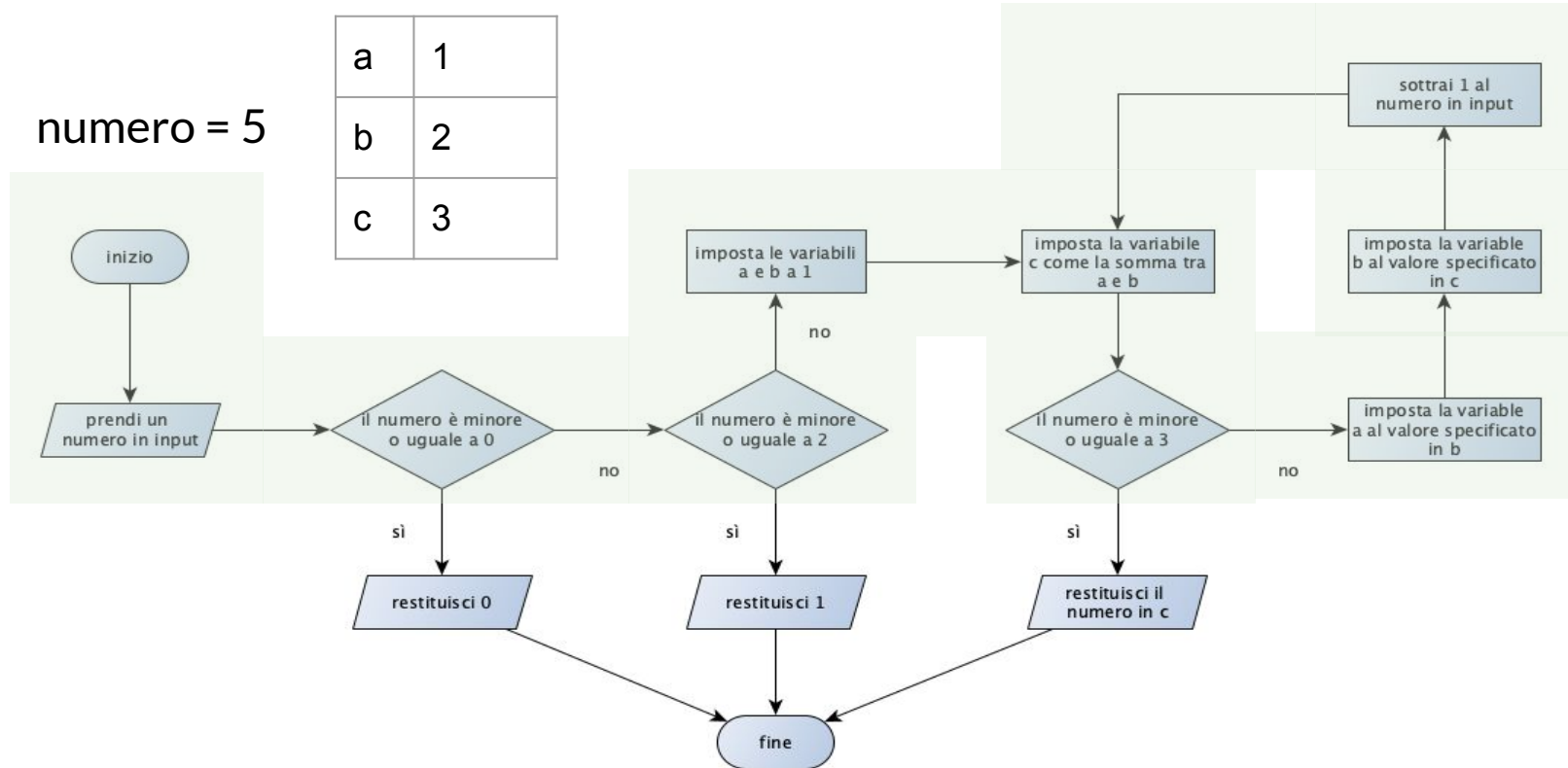


Diagramma di flusso fib(n) per n = 6

numero = 4

a	2
b	3
c	5

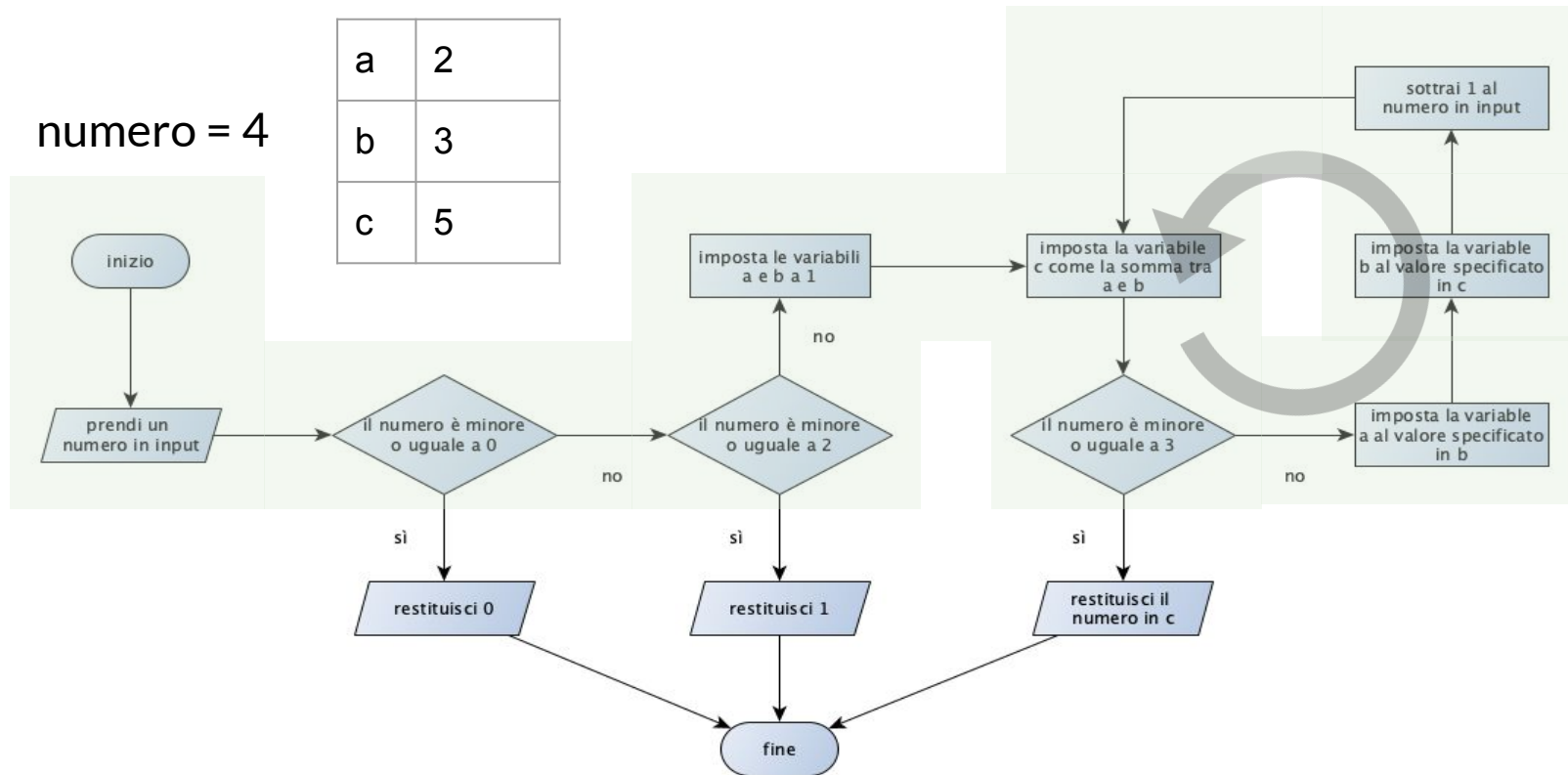


Diagramma di flusso fib(n) per n = 6

numero = 3

a	3
b	5
c	8

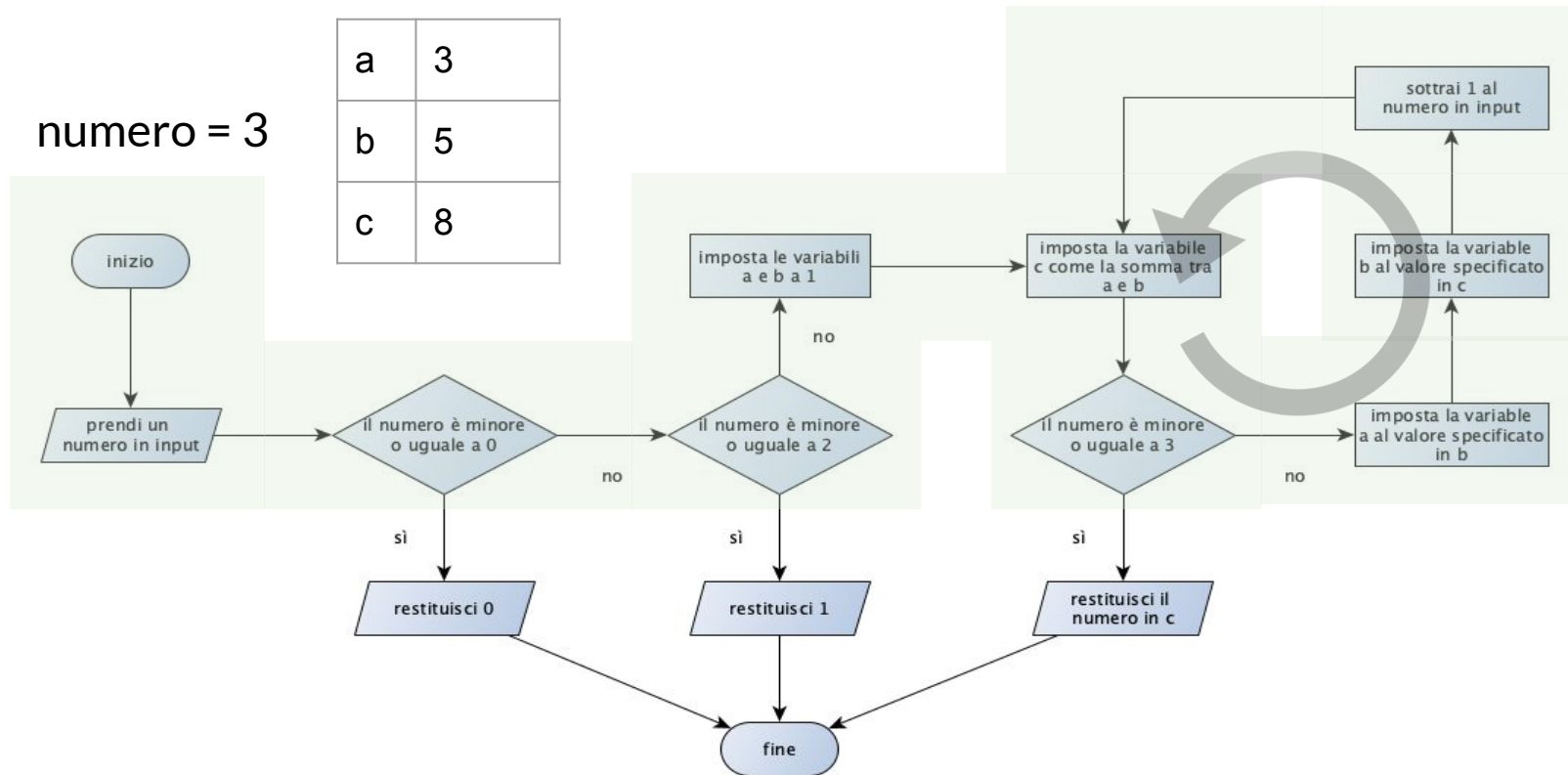


Diagramma di flusso fib(n) per n = 6

numero = 3

a	3
b	5
c	8

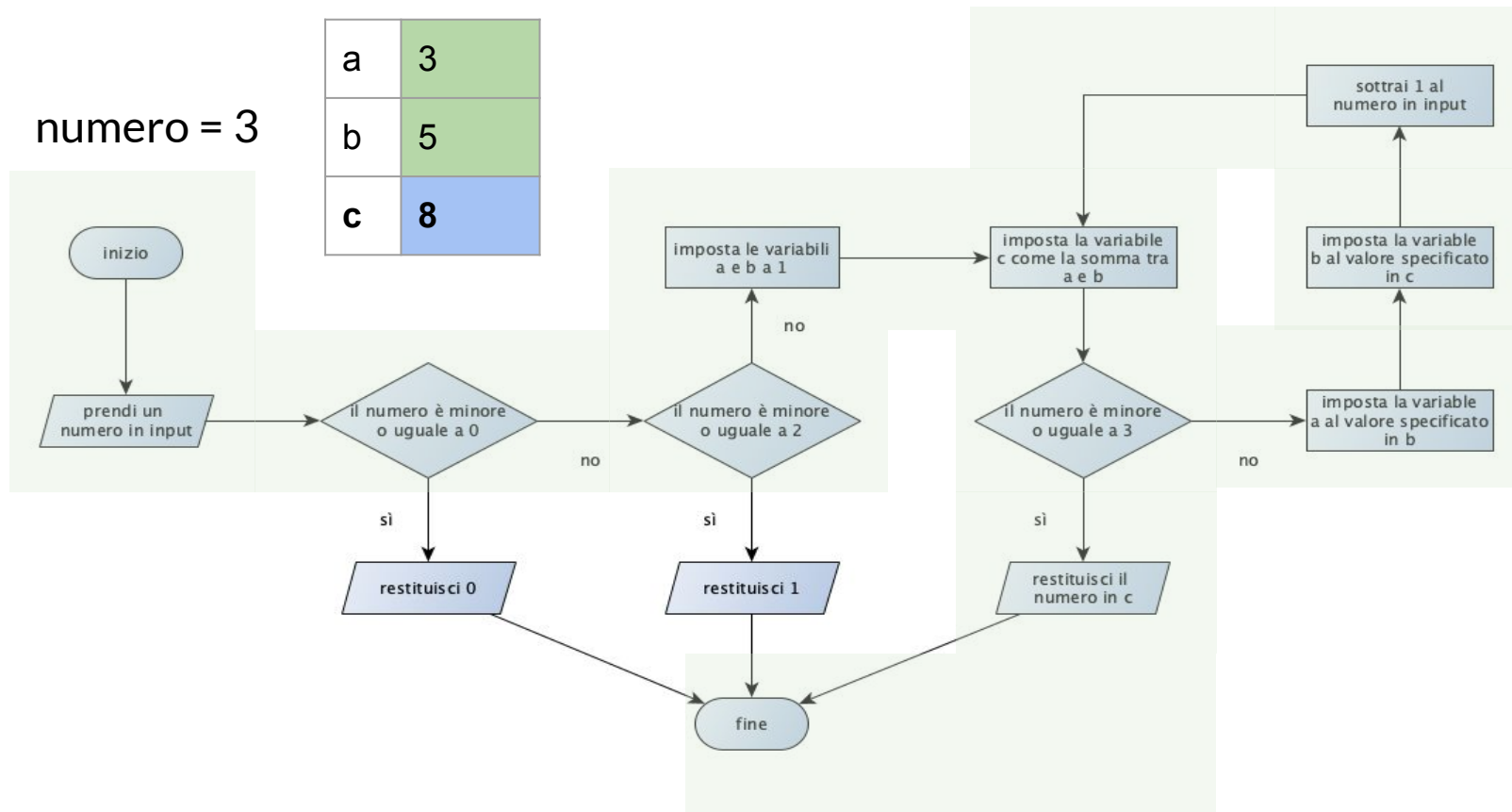
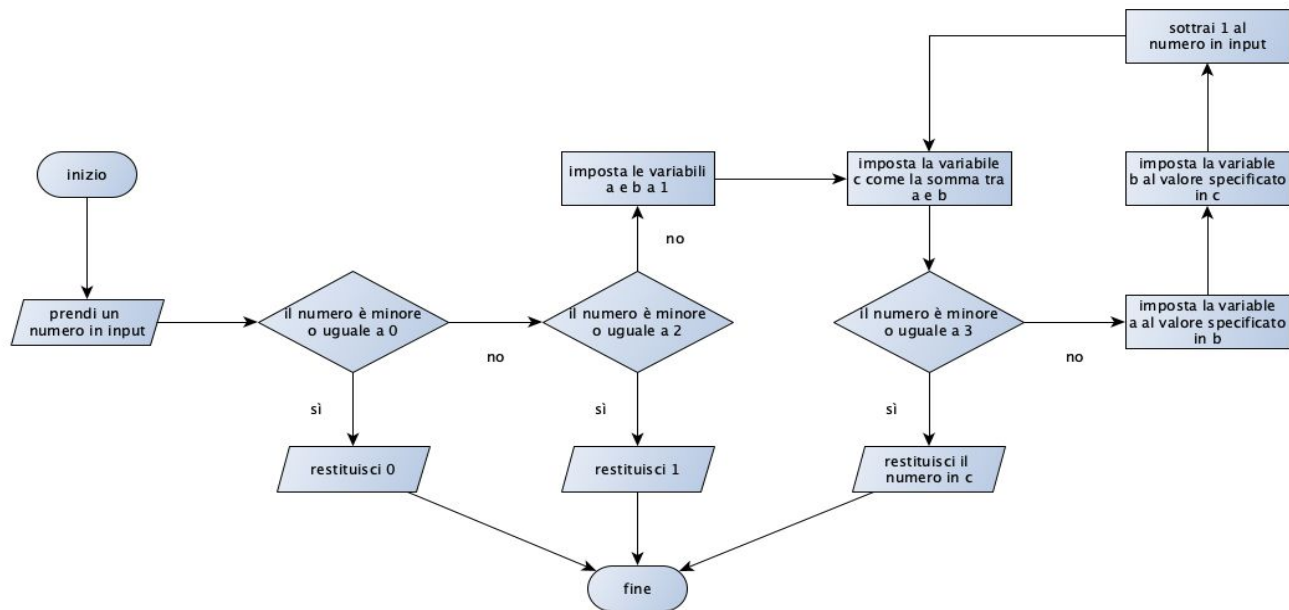


Diagramma di flusso di fib(n) – Python

```
def fib(n):  
    if n <= 0:  
        return 0  
    elif n <= 2:  
        return 1  
    else:  
        a = 1  
        b = 1  
        while True:  
            c = a + b  
            if n <= 3:  
                return c  
            a = b  
            b = c  
            n = n - 1
```



Formalizzazione di un linguaggio

<NOME>: Non Terminale

"VALORE": Terminale

premessa ::= espressione

<frase> ::= <persona> <verbo>

<persona> ::= <nome> <cognome>

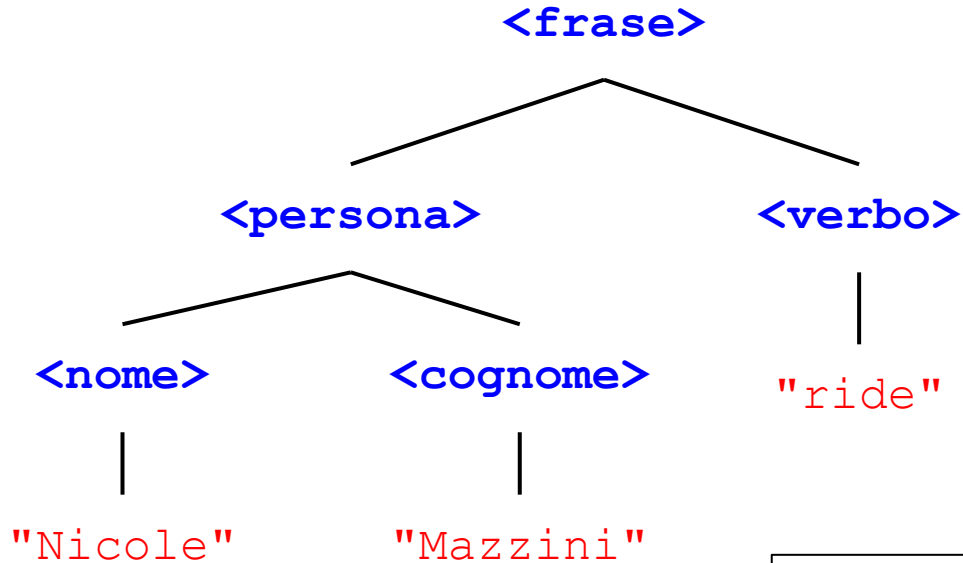
<nome> ::= "Mario" | "Valentina" | "Nicole"

<cognome> ::= "Rossi" | "Mazzini"

<verbo> ::= "ride" | "vola"

Nicole Mazzini ride

Nicole Mazzini ride

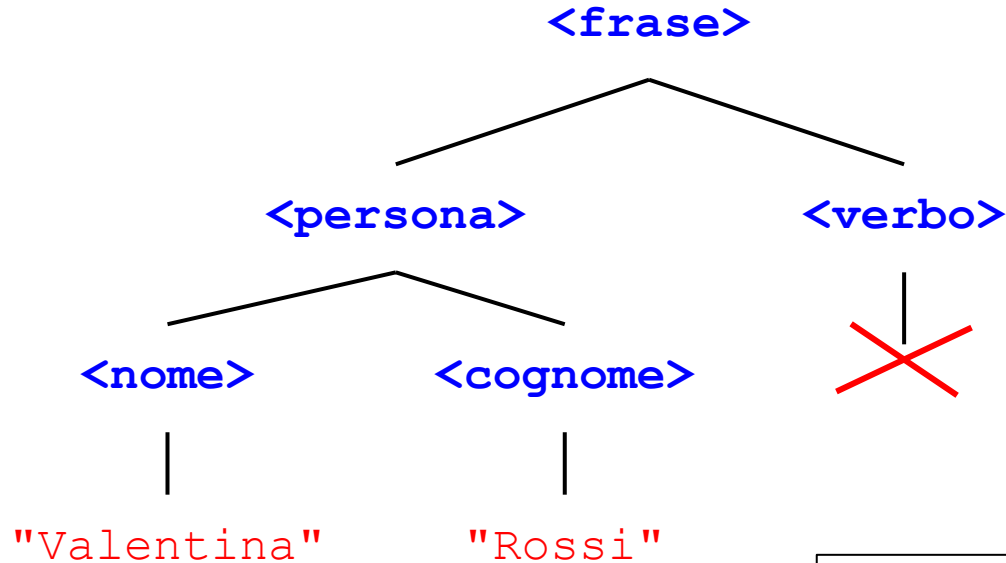


```
<frase> ::= <persona> <verbo>
<persona> ::= <nome> <cognome>
<nome> ::= "Mario" | "Valentina" | "Nicole"
<cognome> ::= "Rossi" | "Mazzini"
<verbo> ::= "ride" | "vola"
```

Valentina Rossi parla

~~Valentina Rossi parla~~

Valentina Rossi parla



```
<frase> ::= <persona> <verbo>
<persona> ::= <nome> <cognome>
<nome> ::= "Mario" | "Valentina" | "Nicole"
<cognome> ::= "Rossi" | "Mazzini"
<verbo> ::= "ride" | "vola"
```

Grammatiche di Chomsky

Chomsky ha proposto una gerarchia per descrivere formalmente le relazioni che possono esistere tra diverse grammatiche in termini delle loro possibili strutture sintattiche che sono in grado di generare

Queste tipologie sono caratterizzate dal tipo di simboli che possono essere usati nella premessa e nell'espressione delle regole di produzione (lettere greche usate per indicare una combinazione di simboli terminali e/o non terminali):

- grammatiche **regolari** - $\langle nt \rangle ::= "t" \mid "t" \langle nt \rangle$
- grammatiche **libere dal contesto** - $\langle nt \rangle ::= \gamma$
- grammatiche **dipendenti dal contesto** - $\alpha \langle nt \rangle \beta ::= \alpha \gamma \beta$
- grammatiche **ricorsivamente enumerabili** - $\alpha ::= \beta$
