

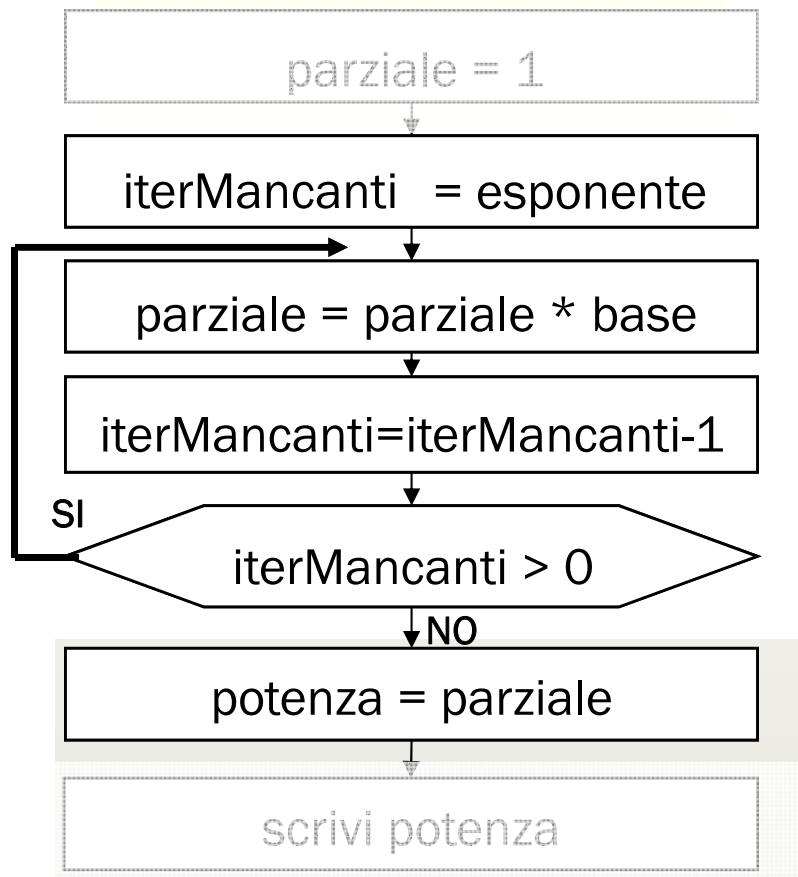


POLITECNICO
DI MILANO

INFORMATICA

Dal diagramma di
flusso
al programma in C++

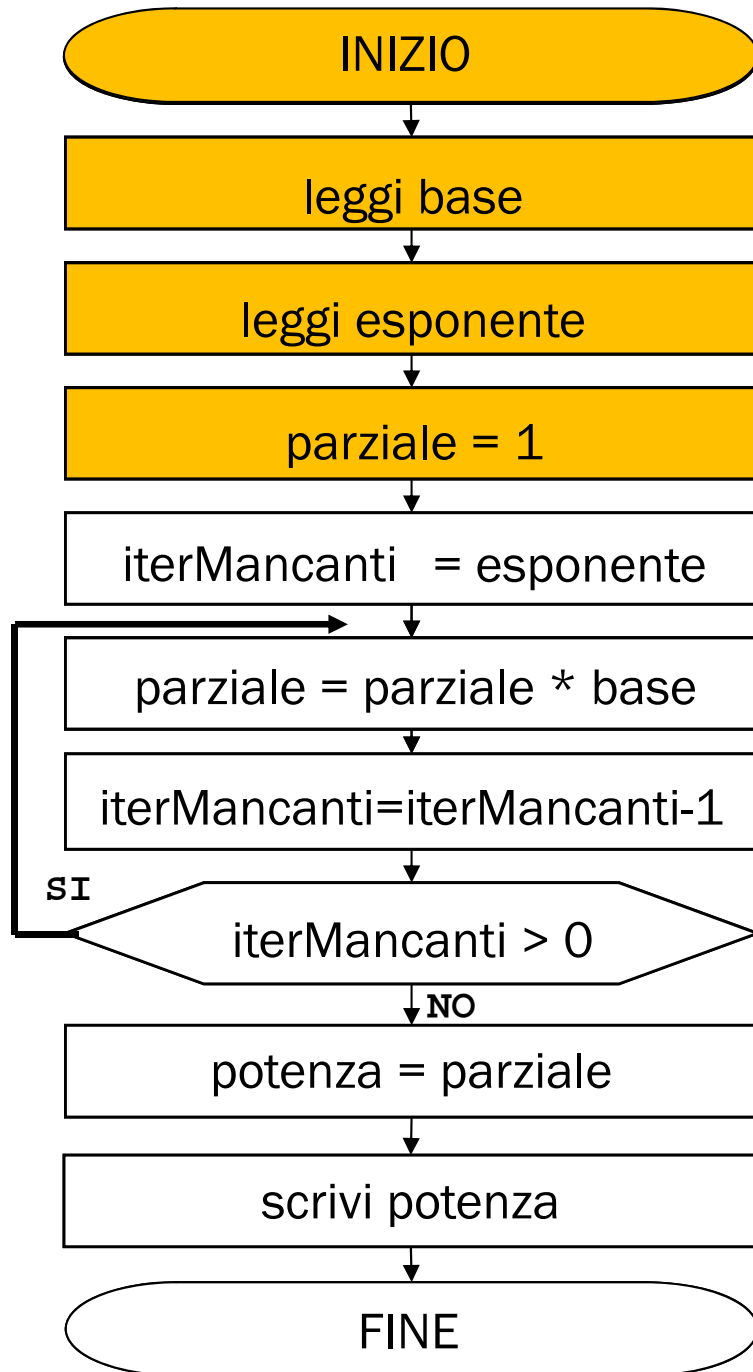
Forma grafica

**CICLO**

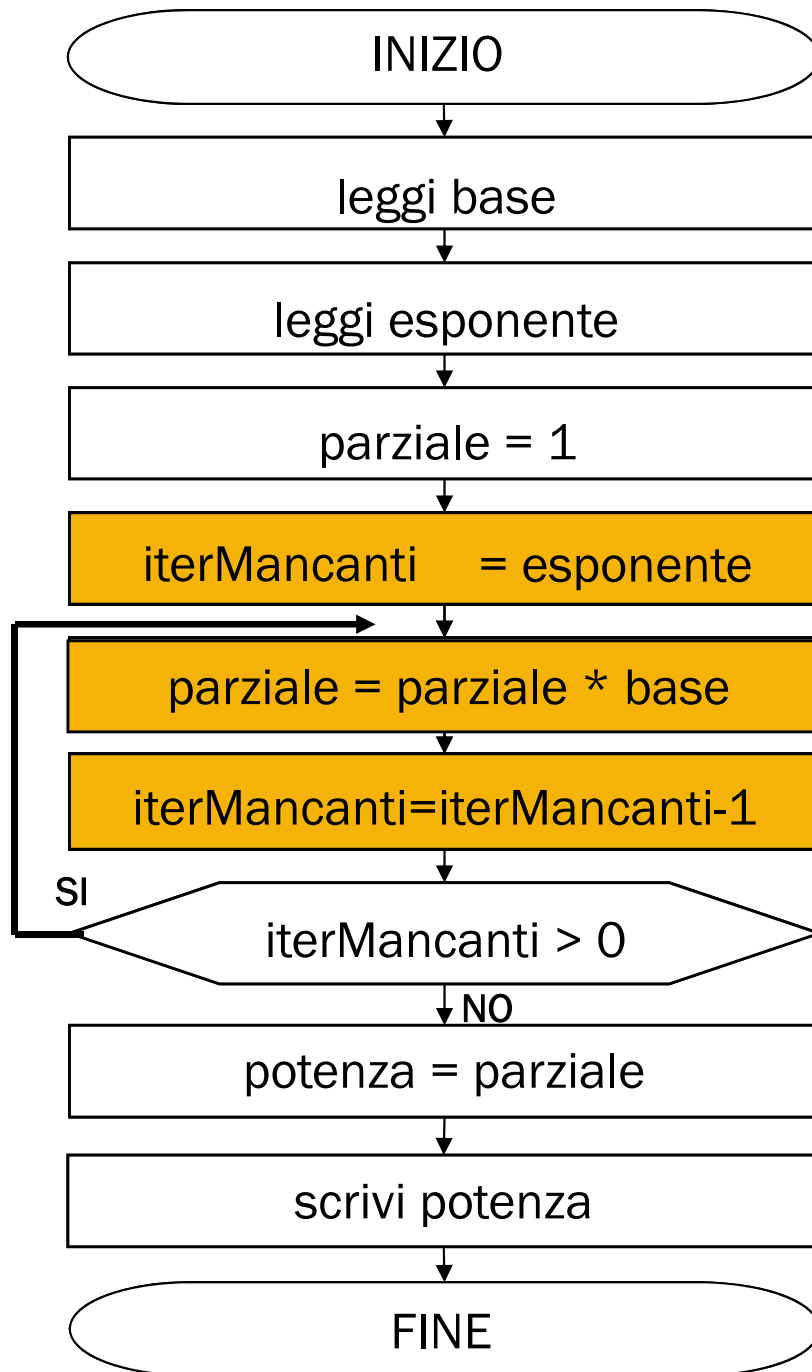
Forma testuale

```
parziale = 1
iterMancanti = esponente
parziale = parziale * base
iterMancanti = iterMancanti - 1
se iterMancanti > 0 vai a CICLO
potenza = parziale
scrivi potenza
```

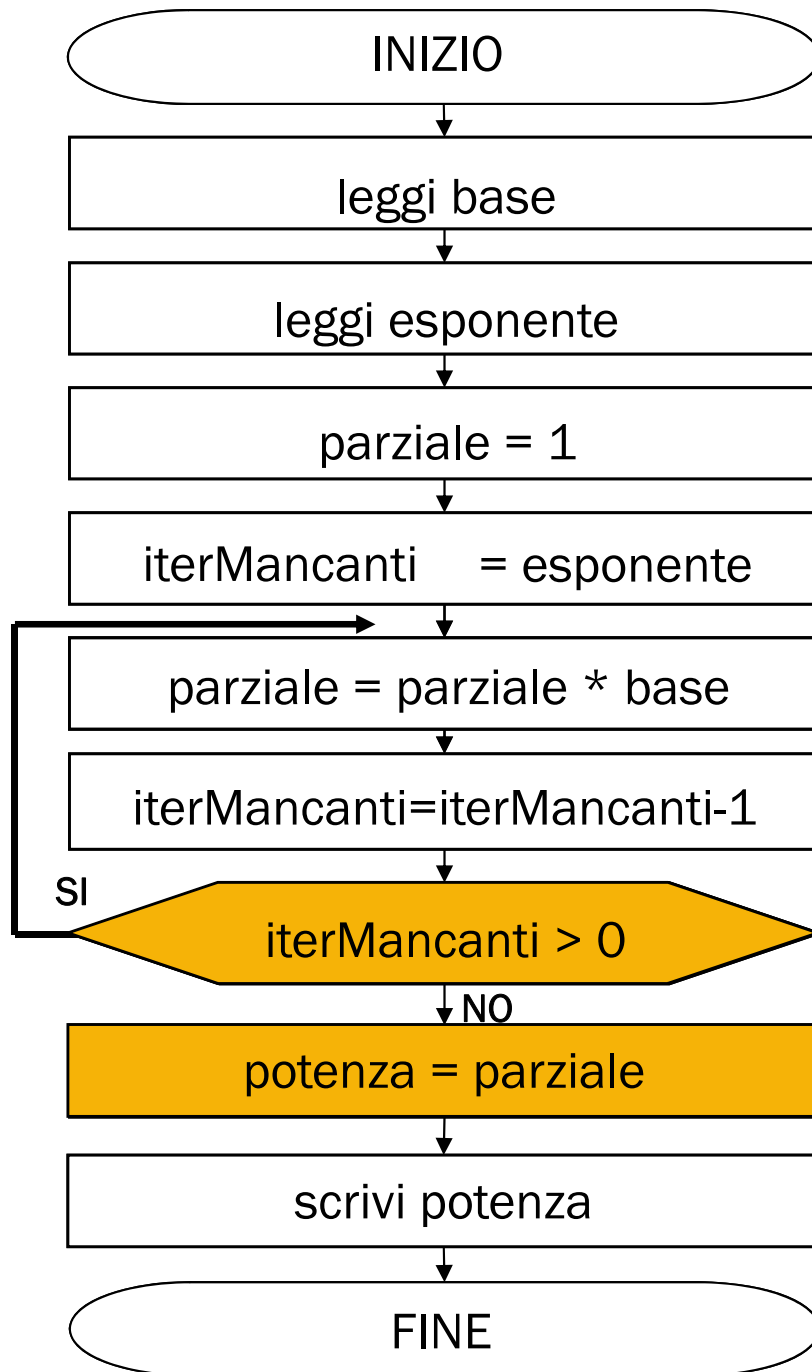
RIGHE SUCCESSIVE DI
TESTO



leggi base
leggi esponente
parziale = 1



leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
parziale = parziale * base
iterMancanti = iterMancanti - 1



leggi base

leggi esponente

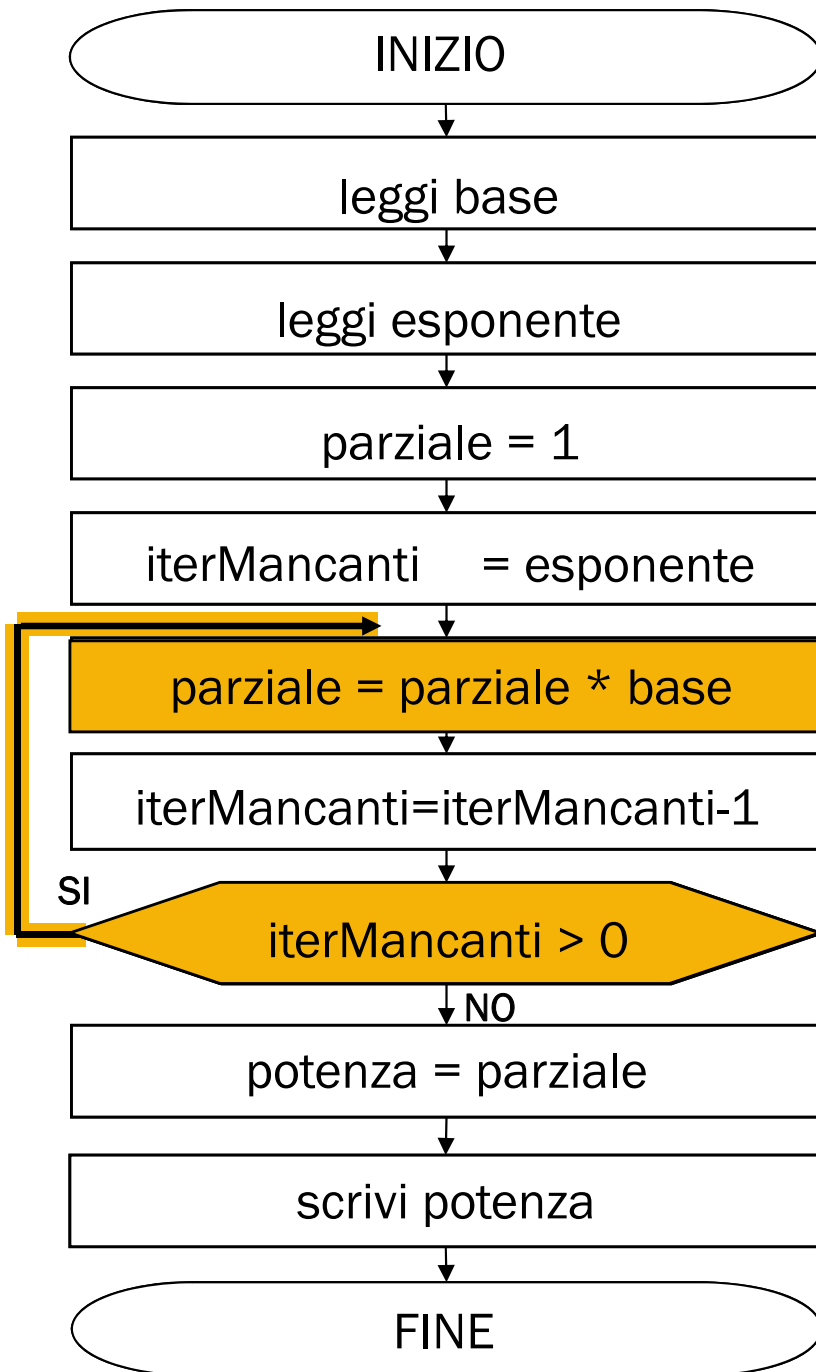
parziale = 1

iterMancanti = esponente

parziale = parziale * base

iterMancanti = iterMancanti - 1

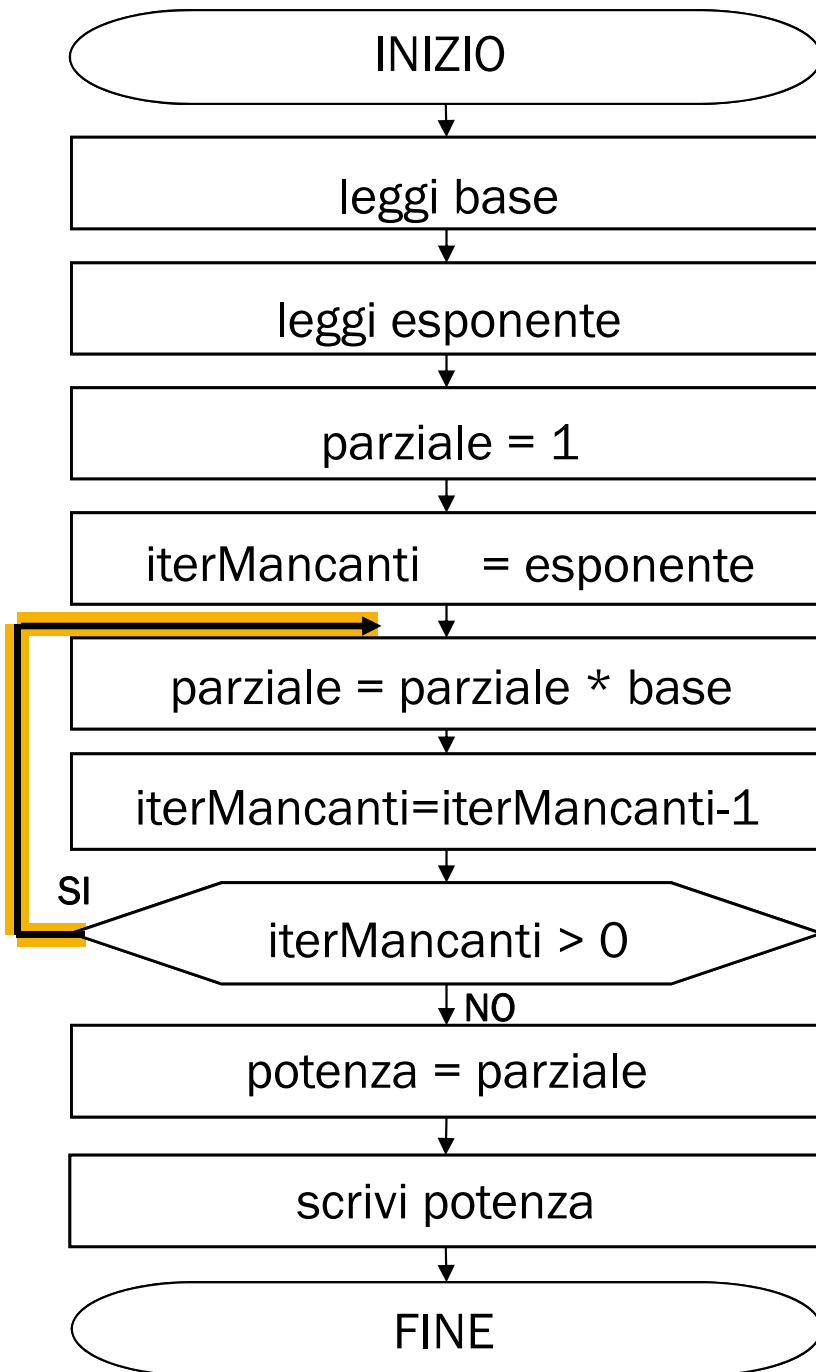
potenza = parziale



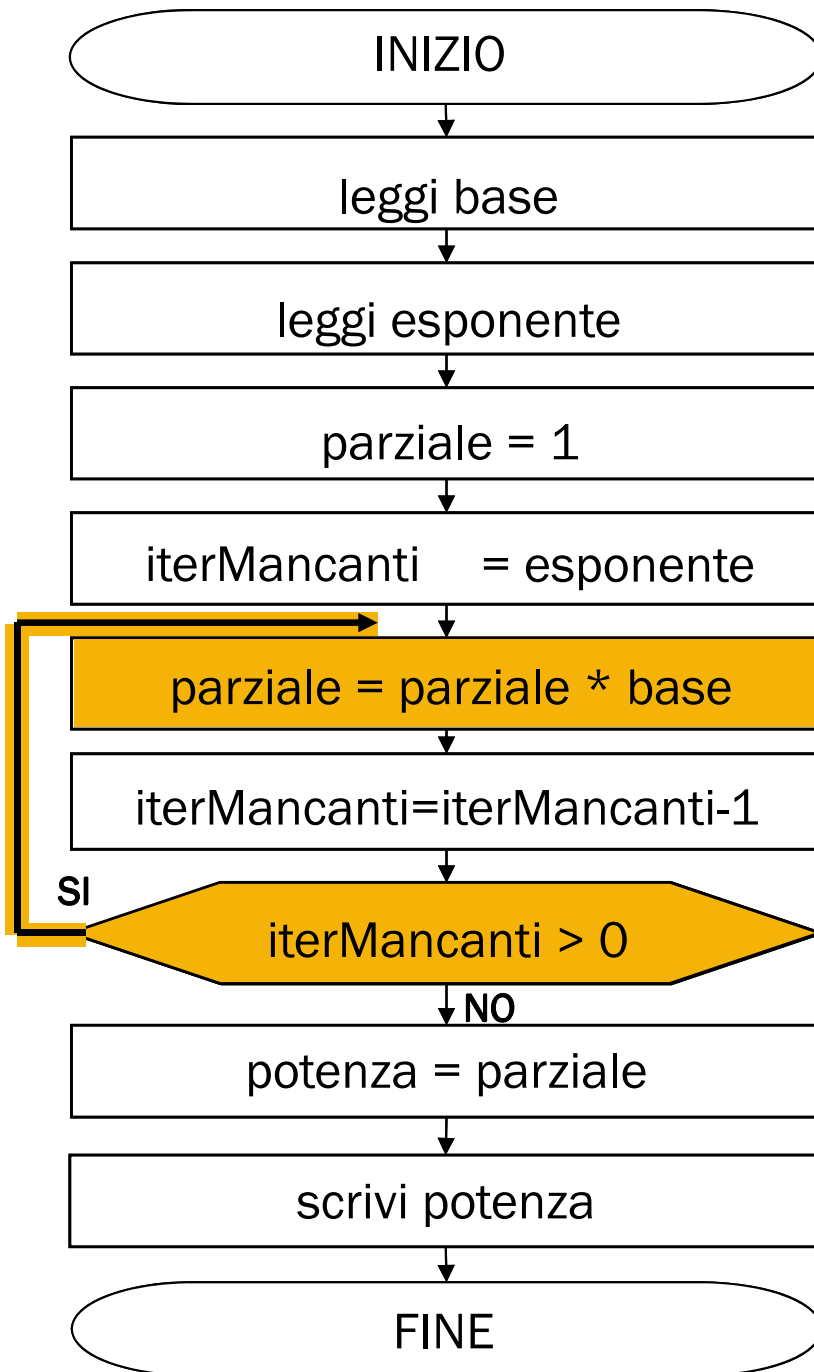
ECCEZIONE

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
parziale = parziale * base
iterMancanti = iterMancanti - 1

potenza = parziale
```

ISTRUZIONE DI
SALTO

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
parziale = parziale * base
iterMancanti = iterMancanti - 1
se iterMancanti > 0 vai a...
potenza = parziale
```



leggi base

leggi esponente

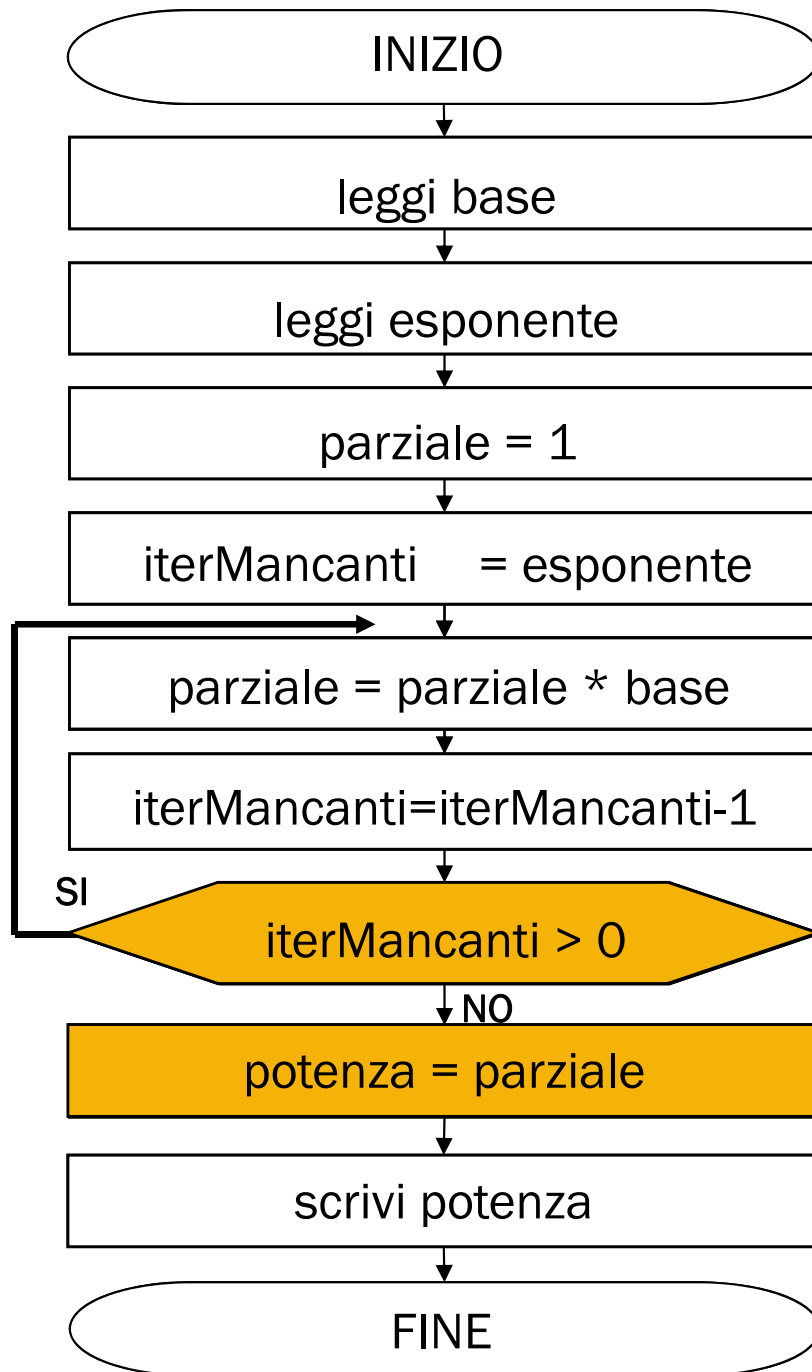
parziale = 1

iterMancanti = esponente

parziale = parziale * base

iterMancanti = iterMancanti - 1

potenza = parziale

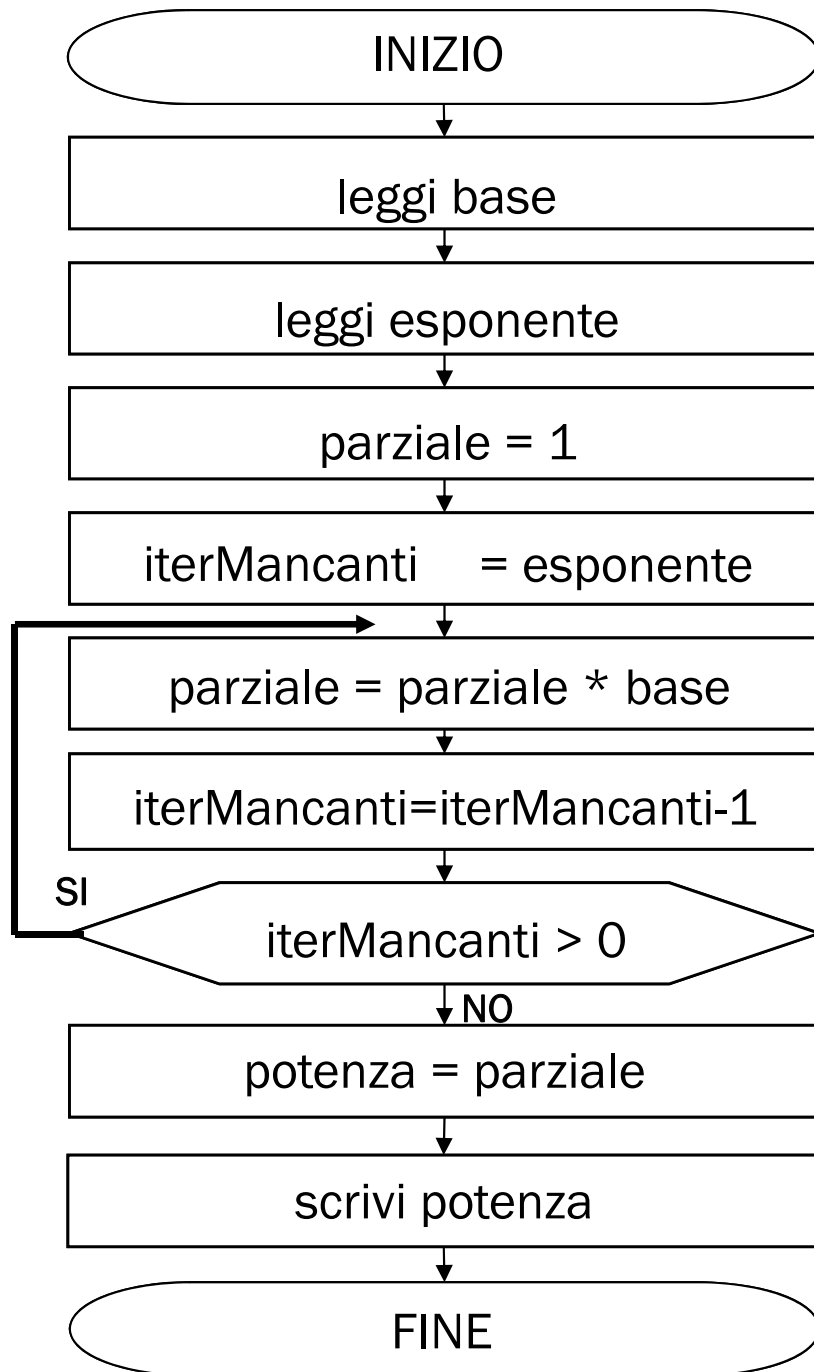


REGOLA GENERALE

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
parziale = parziale * base
iterMancanti = iterMancanti - 1
```

potenza = parziale

ETICHETTA



CICLO

leggi base

leggi esponente

parziale = 1

iterMancanti = esponente

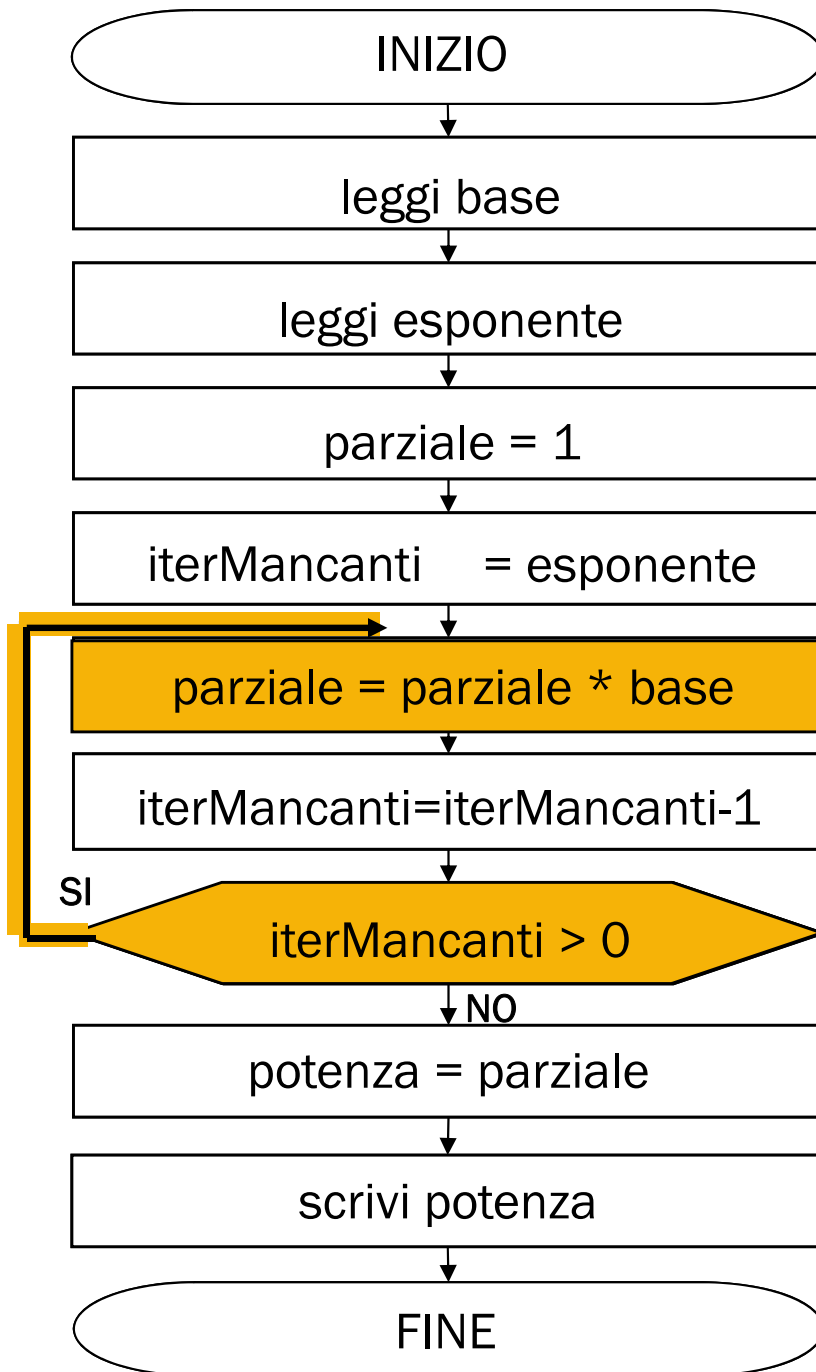
parziale = parziale * base

iterMancanti = iterMancanti - 1

se iterMancanti > 0 vai a

CICLO

potenza = parziale



leggi base

leggi esponente

parziale = 1

iterMancanti = esponente

CICLO parziale = parziale * base

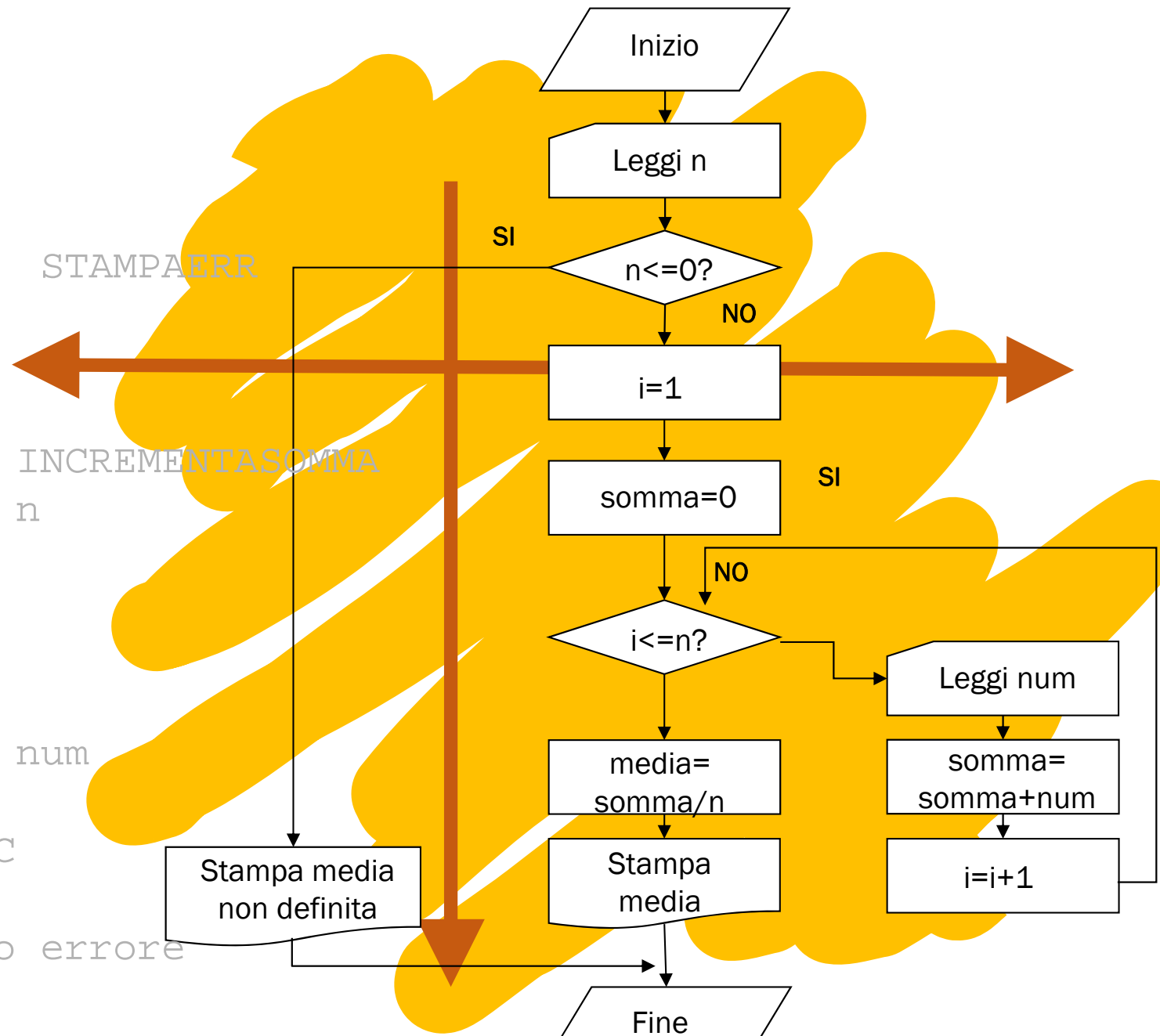
iterMancanti = iterMancanti - 1

se iterMancanti > 0 **vai** a **CICLO**

potenza = parziale

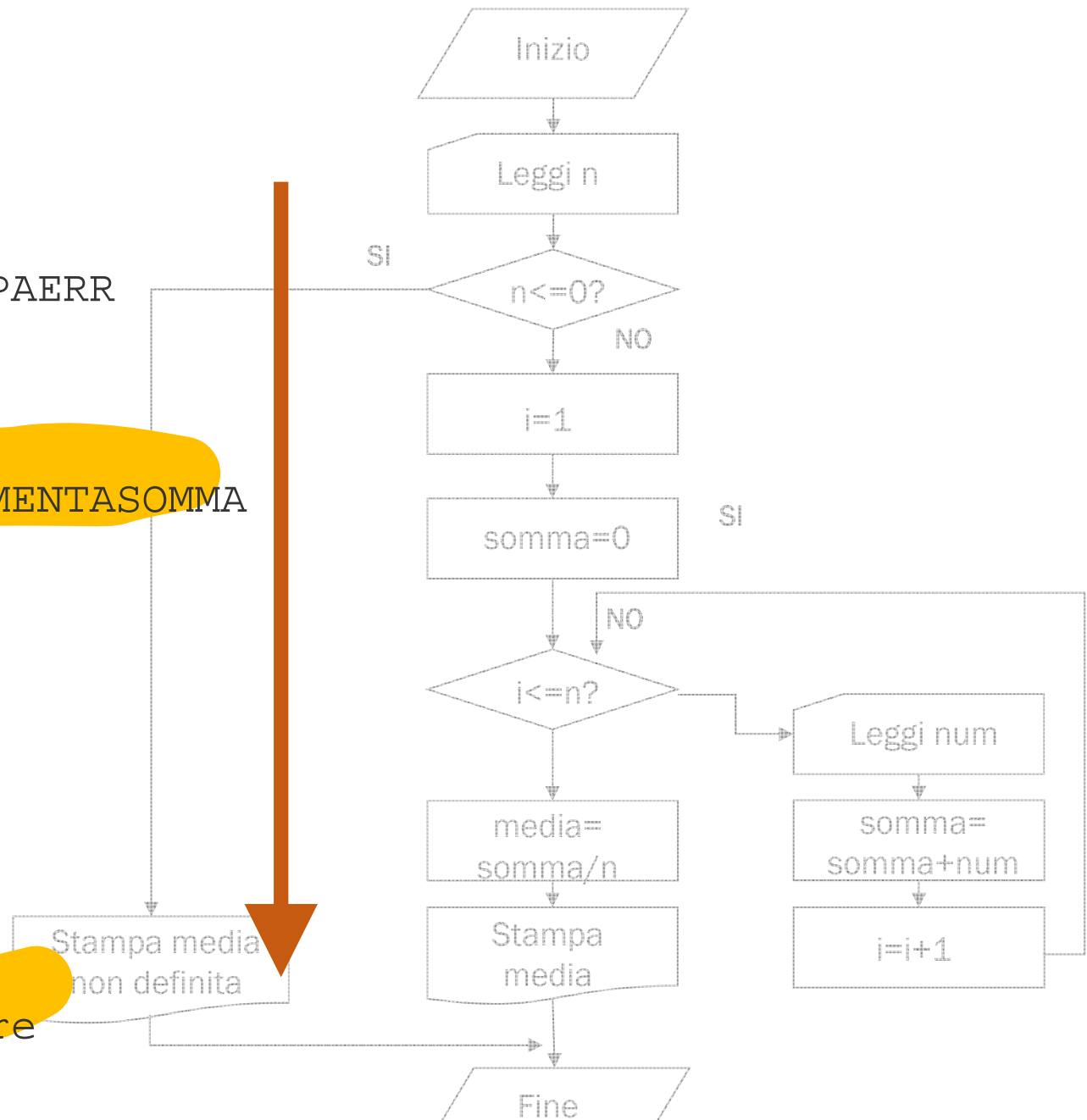
BIDIMENSIONALE

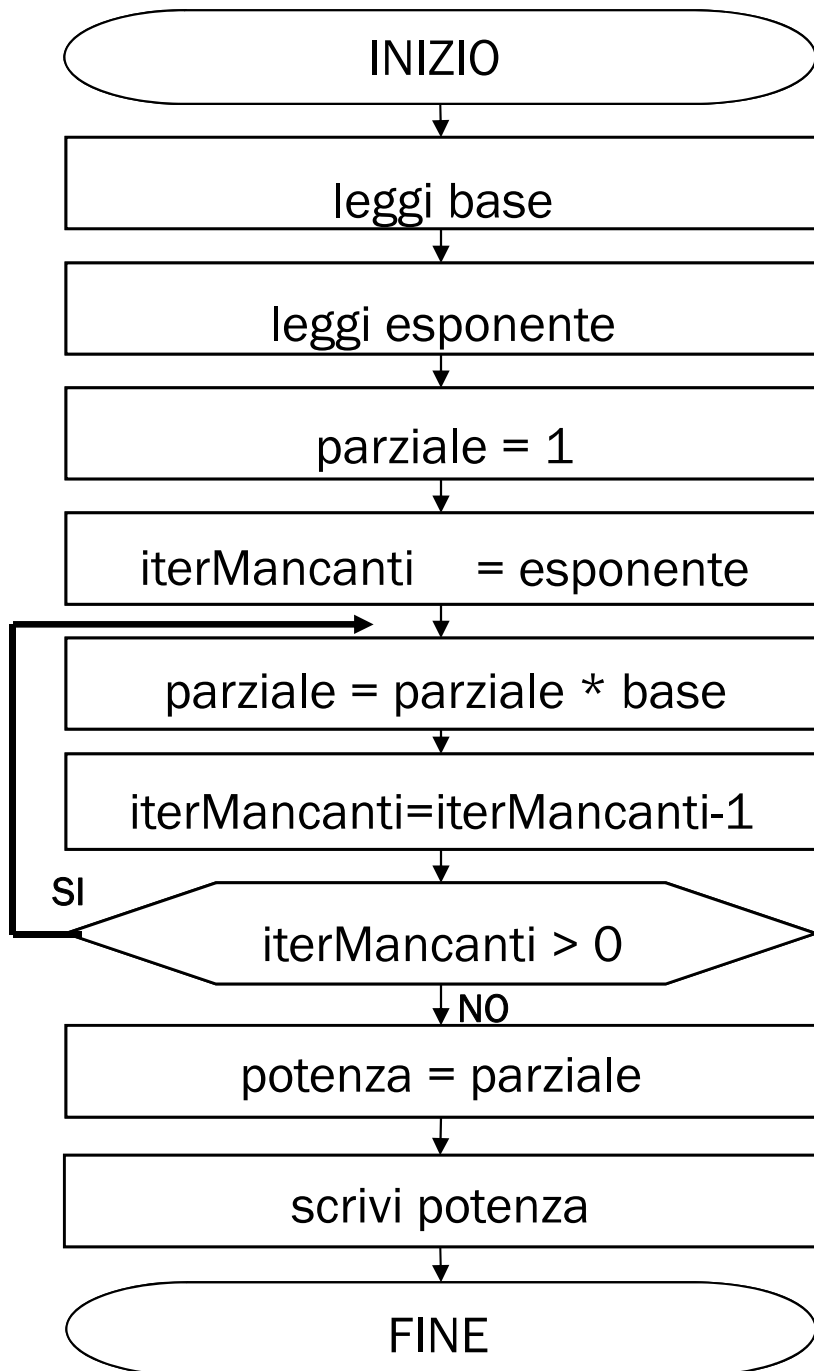
```
leggi n
se n <= 0 vai a STAMPAERR
i = 1
somma = 0
CICLOPRINC
se 1 <= n vai a INCREMENTASOMMA
media = somma / n
stampa media
vai a FINE
INCREMENTASOMMA
Leggi num
somma = somma + num
i = i + 1
vai a CICLOPRINC
STAMPAERR
stampa messaggio errore
FINE
```



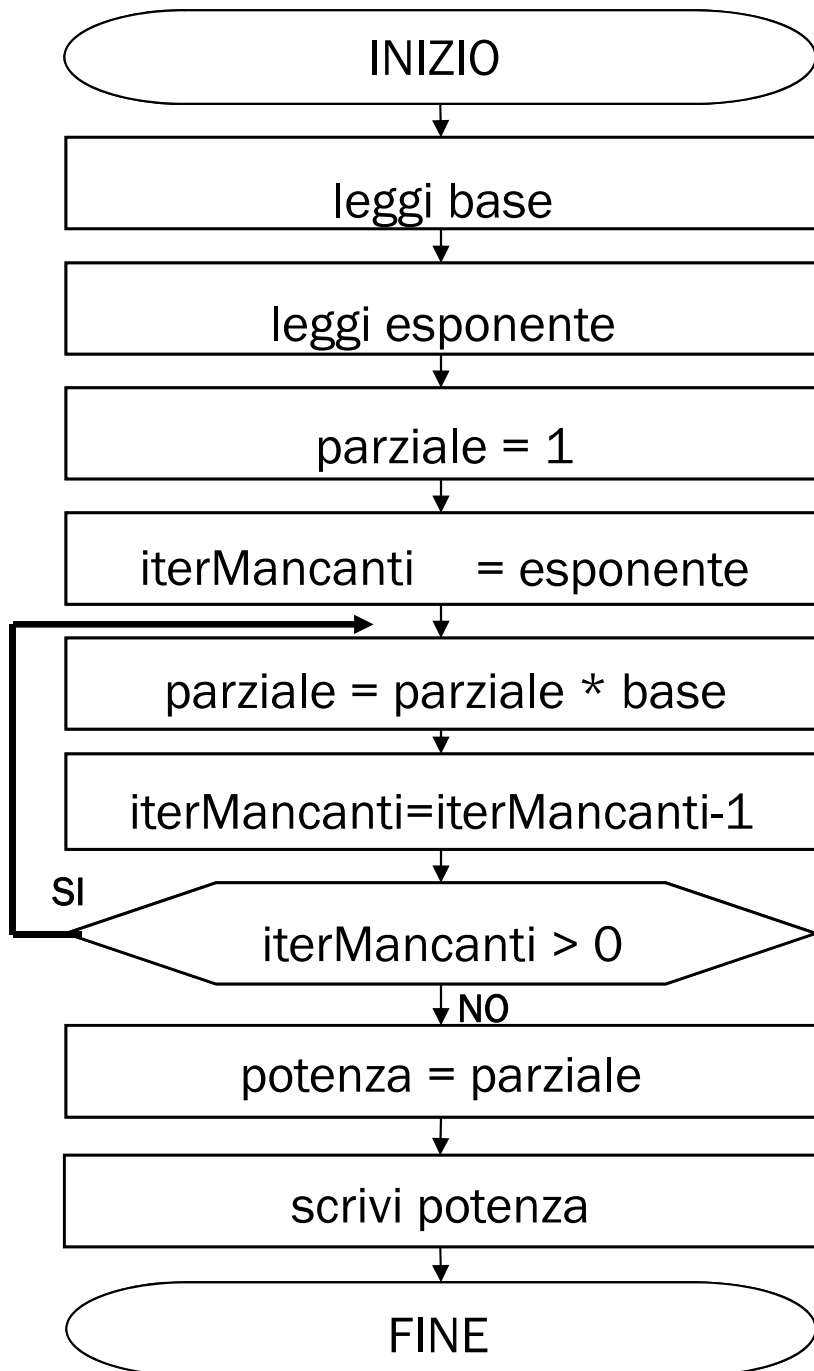
MONODIMENSIONALE

```
leggi n
se n <= 0 vai a STAMPAERR
i = 1
somma = 0
CICLOPRINC
se 1 <= n vai a INCREMENTASOMMA
media = somma / n
stampa media
vai a FINE
INCREMENTASOMMA
Leggi num
somma = somma + num
i = i + 1
vai a CICLOPRINC
STAMPAERR
stampa messaggio errore
FINE
```

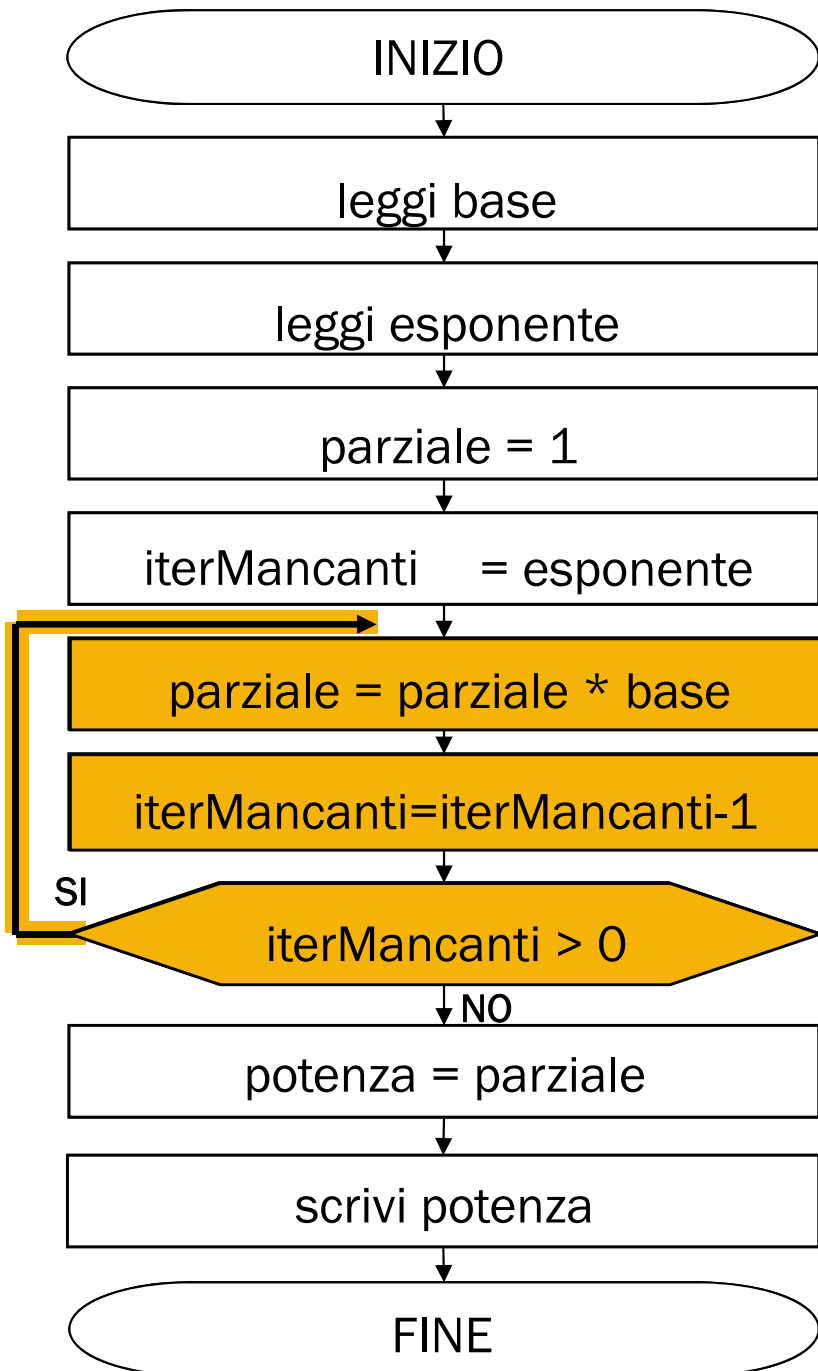


**CICLO**

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
CICLO
parziale = parziale * base
iterMancanti = iterMancanti - 1
se iterMancanti > 0 vai a CICLO
potenza = parziale
scrivi potenza
fine
```



```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
esegui
parziale = parziale * base
iterMancanti = iterMancanti - 1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```

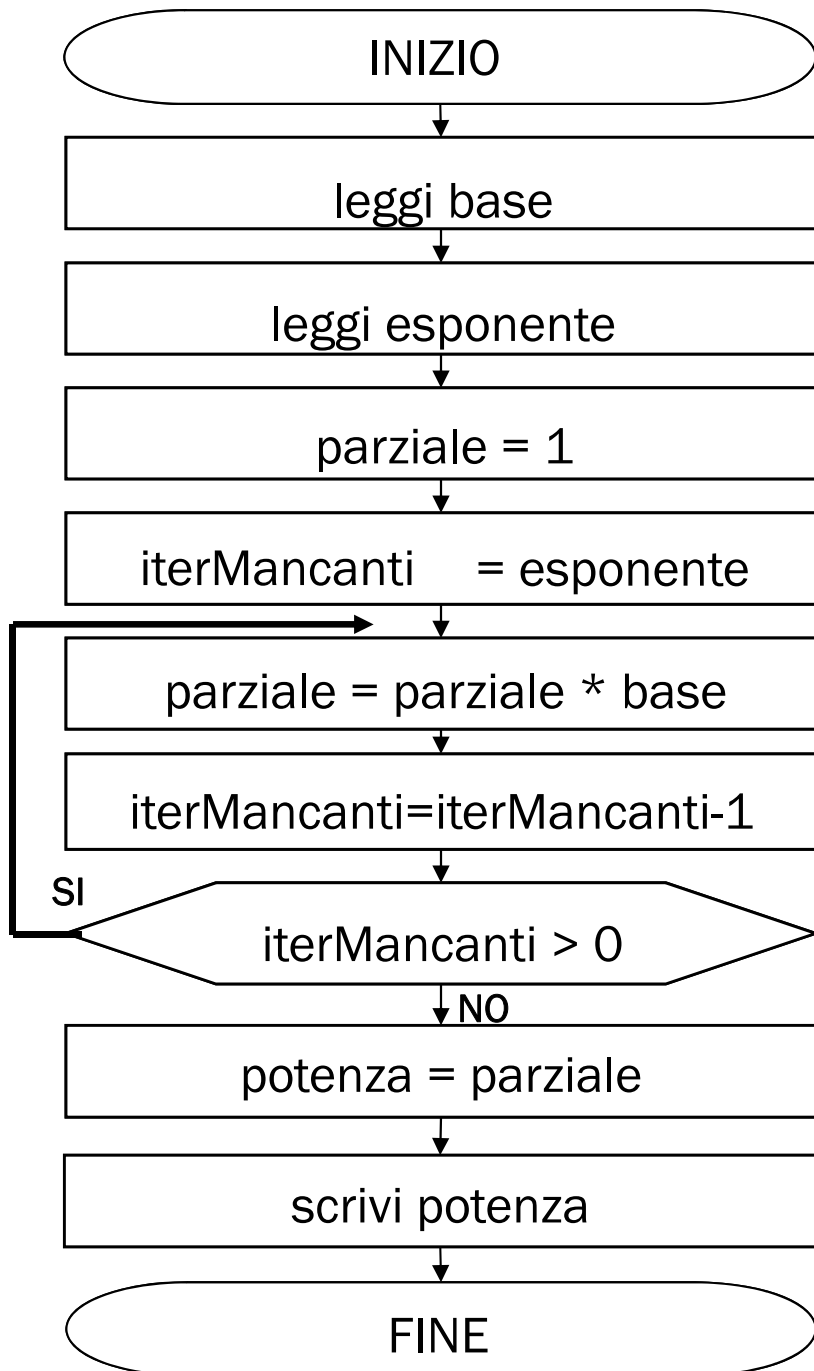


```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
```

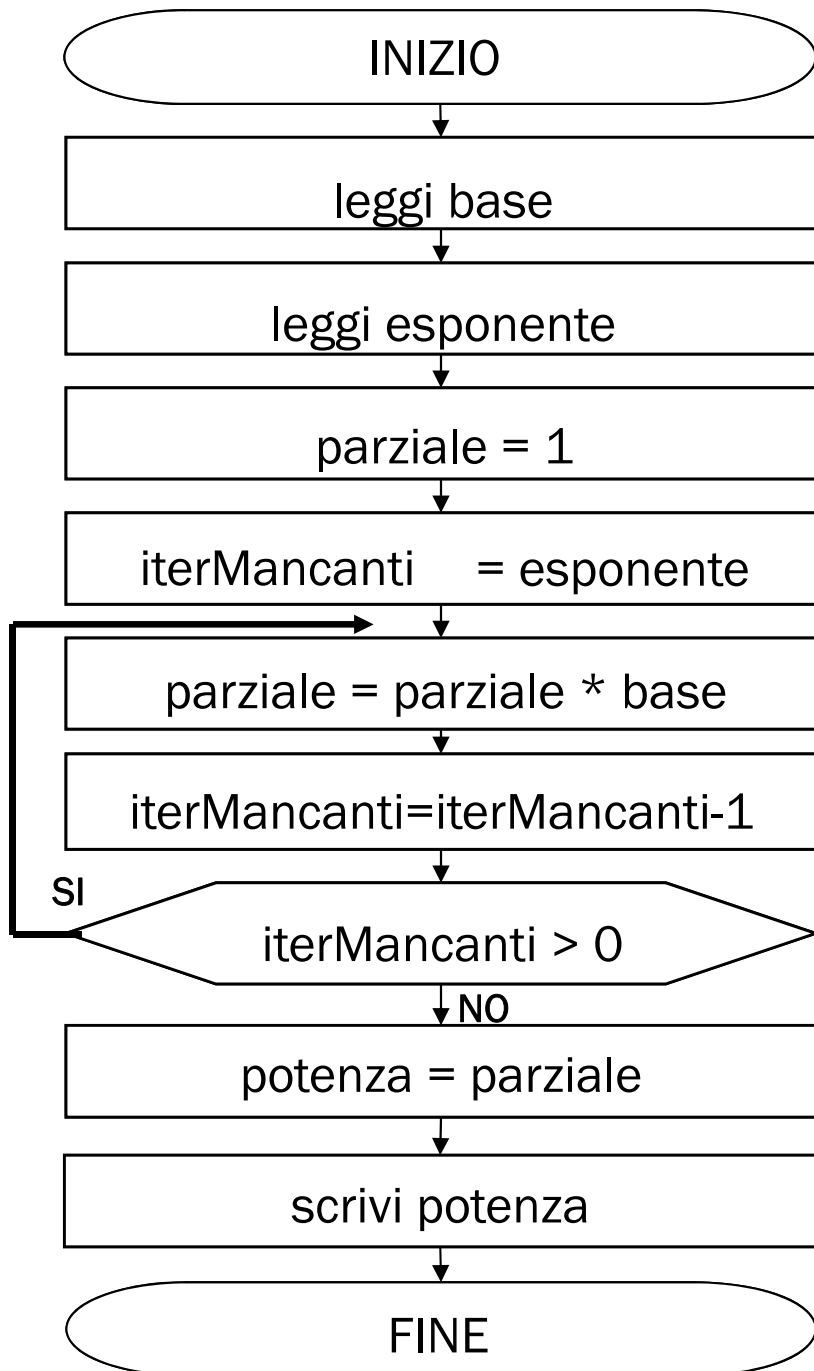
esegui

```
parziale = parziale * base
iterMancanti = iterMancanti - 1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```


STRUTTURE DI CONTROLLO



```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
esegui
parziale = parziale * base
iterMancanti = iterMancanti - 1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```



PAROLE CHIAVE

`leggi base``leggi esponente``parziale = 1``iterMancanti = esponente`**`esegui`**`parziale = parziale * base``iterMancanti = iterMancanti - 1`**`fintantoché`** `iterMancanti > 0``potenza = parziale``scrivi potenza``fine`

1

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
CICLO parziale = parziale * base
iterMancanti = iterMancanti -1
se iterMancanti > 0 vai a CICLO
potenza = parziale
scrivi potenza
fine
```

2

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
esegui
parziale = parziale * base
iterMancanti = iterMancanti -1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
CICLO parziale = parziale * base
      iterMancanti = iterMancanti -1
      se iterMancanti > 0 vai a CICLO
potenza = parziale
scrivi potenza
fine
```

PASCAL, C, C++

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
esegui
parziale = parziale * base
iterMancanti = iterMancanti -1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
CICLO parziale = parziale * base
      iterMancanti = iterMancanti -1
      se iterMancanti > 0 vai a CICLO
potenza = parziale
scrivi potenza
fine
```

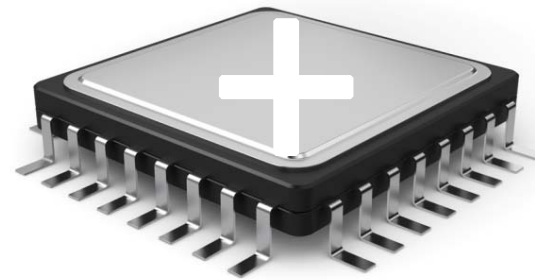
STRUTTURA
LEGGIBILE

```
leggi base
leggi esponente
parziale = 1
iterMancanti = esponente
esegui
parziale = parziale * base
iterMancanti = iterMancanti -1
fintantoché iterMancanti > 0
potenza = parziale
scrivi potenza
fine
```

STRUTTURA
LEGGIBILE

E' importante rendere il programma
più leggibile e, quindi,
più facilmente modificabile

SUPPONIAMO:



CICLO

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

[contatore] = base
[variabile] = 0
[ETICH] [variabile] = [variabile] + parziale
[contatore] = [contatore] - 1
se [contatore] > 0 vai a [ETICH]
parziale = [variabile]


```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

[contatore] = base
[variabile] = 0
[ETICH] **[variabile] = [variabile] + parziale**
[contatore] = [contatore] - 1
se [contatore] > 0 vai a [ETICH]
parziale = [variabile]

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

[contatore] = base
[variabile] = 0
[ETICH] [variabile] = [variabile] + parziale
[contatore] = [contatore] - 1
se [contatore] > 0 vai a [ETICH]
parziale = [variabile]

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

[contatore] = base
[variabile] = 0
[ETICH] [variabile] = [variabile] + parziale
[contatore] = [contatore] - 1
se [contatore] > 0 vai a [ETICH]
parziale = [variabile]

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

[contatore] = base
prodParziale = 0
[ETICH] prodParziale = prodParziale + parziale
contatore] = [contatore] - 1
se [contatore] > 0 vai a [ETICH]
parziale = prodParziale

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

```
[contatore] = base  
prodParziale = 0
```

```
[ETICH] prodParziale = prodParziale + parziale  
[contatore] = [contatore] - 1  
se [contatore] > 0 vai a [ETICH]  
parziale = prodParziale
```

```
nti = esponente  
= = parziale * base  
nti = iterMancanti  
ncanti > 0
```

```
[contatore] = base  
prodParziale = 0
```

```
[ETICH] prodParziale = prodParziale + potParziale  
[contatore] = [contatore] - 1  
se [contatore] > 0 vai a [ETICH]  
potParziale = prodParziale
```

```
nti = esponente
```

```
le = parziale * base
```

```
nti = iterMancanti
```

```
ncanti > 0
```

[contatore] = base

prodParziale = 0

[ETICH] prodParziale = prodParziale + parziale

[contatore] = **[contatore]** - 1

se **[contatore]** > 0 vai a [ETICH]

parziale = prodParziale

```
nti = esponente
```

```
le = parziale * base
```

```
nti = iterMancanti
```

```
ncanti > 0
```

somMancanti = base

prodParziale = 0

[ETICH] prodParziale = prodParziale + potParziale

somMancanti = **somMancanti** - 1

se **somMancanti** > 0 vai a [ETICH]

potParziale = prodParziale

```
leggi base  
leggi esponente  
potParziale = 1  
iterMancanti = esponente
```

CICLO

```
somMancanti = base  
prodParziale = 0
```

```
[ETICH] prodParziale = prodParziale + potParziale  
        somMancanti = somMancanti - 1  
        se somMancanti > 0 vai a [ETICH]  
        potParziale = prodParziale  
        iterMancanti = iterMancanti - 1  
        se iterMancanti > 0 vai a CICLO  
        potenza = potParziale  
        scrivi potenza  
fine
```

```
leggi base  
leggi esponente  
potParziale = 1  
prodMancanti = esponente
```

CICLO

```
somMancanti = base  
prodParziale = 0
```

```
[ETICH] prodParziale = prodParziale + potParziale  
        somMancanti = somMancanti - 1  
        se somMancanti > 0 vai a [ETICH]  
        prodParziale = prodParziale  
        prodMancanti = prodMancanti - 1  
        se prodMancanti > 0 vai a CICLO  
        potenza = potParziale  
        scrivi potenza  
        fine
```

```
leggi base  
leggi esponente  
potParziale = 1  
prodMancanti = esponente
```

CICLO

```
somMancanti = base  
prodParziale = 0
```

```
[ETICH] prodParziale = prodParziale + potParziale  
somMancanti = somMancanti - 1  
se somMancanti > 0 vai a [ETICH]  
potParziale = prodParziale  
prodMancanti = prodMancanti - 1  
se prodMancanti > 0 vai a CICLO  
potenza = potParziale  
scrivi potenza  
fine
```

```
leggi base  
leggi esponente  
potParziale = 1  
prodMancanti = esponente
```

CICLO

```
somMancanti = base  
prodParziale = 0
```

```
CICLOPROD prodParziale = prodParziale + potParziale  
            somMancanti = somMancanti - 1  
            se somMancanti > 0 vai a CICLOPROD  
            prodParziale = prodParziale  
            prodMancanti = prodMancanti - 1  
            se prodMancanti > 0 vai a CICLO  
            potenza = potParziale  
            scrivi potenza  
            fine
```



```
leggi base  
leggi esponente  
potParziale = 1  
prodMancanti = esponente
```

CICLO

```
somMancanti = base  
prodParziale = 0
```

```
CICLOPROD prodParziale = prodParziale + potParziale  
           somMancanti = somMancanti - 1  
           se somMancanti > 0 vai a CICLOPROD  
           potParziale = prodParziale  
           prodMancanti = prodMancanti - 1  
           se prodMancanti > 0 vai a CICLO  
           potenza = potParziale  
           scrivi potenza  
           fine
```

```
leggi base  
leggi esponente  
potParziale = 1  
prodMancanti = esponente
```

CICLOPOT

```
somMancanti = base  
prodParziale = 0
```

```
CICLOPROD prodParziale = prodParziale + potParziale  
somMancanti = somMancanti - 1  
se somMancanti > 0 vai a CICLOPROD  
potParziale = prodParziale  
prodMancanti = prodMancanti - 1  
se prodMancanti > 0 vai a CICLOPOT  
potenza = potParziale  
scrivi potenza  
fine
```

```
leggi base
leggi esponente
potParziale = 1
prodMancanti = esponente
esegui
potParziale = potParziale * base
prodMancanti = prodMancanti - 1
fintantochè prodMancanti > 0
potenza = potParziale
scrivi potenza
fine
```

```
leggi base
leggi esponente
potParziale = 1
prodMancanti = esponente
esegui
potParziale = potParziale * base
prodMancanti = prodMancanti - 1
fintantochè prodMancanti > 0
potenza = potParziale
scrivi potenza
fine
```

STRUTTURE DI
CONTROLLO

```
leggi base
leggi esponente
potParziale = 1
prodMancanti = esponente
esegui
  potParziale = potParziale * base
  prodMancanti = prodMancanti - 1
fintantochè prodMancanti > 0
  potenza = potParziale
  scrivi potenza
fine
```

C++ SINTASSI

```
leggi base
leggi esponente
potParziale = 1
prodMancanti = esponente
esegui
potParziale = potParziale * base
prodMancanti = prodMancanti - 1
fintantochè prodMancanti > 0
potenza = potParziale
scrivi potenza
fine
```

CARATTERE
TERMINATORE ;

leggi base;

leggi esponente;

potParziale = 1;

prodMancanti = esponente;

esegui

potParziale = potParziale * base;

prodMancanti = prodMancanti - 1;

fintantoché prodMancanti > 0;

potenza = potParziale;

scrivi potenza;

fine;

CARATTERE
TERMINATORE ;

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
esegui  
potParziale = potParziale * base;  
prodMancanti = prodMancanti - 1;  
fintantoché prodMancanti > 0;  
potenza = potParziale;  
scrivi potenza;  
fine;
```


ISTRUZIONI IN
INGLESE

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
esegui  
potParziale = potParziale * base;  
prodMancanti = prodMancanti - 1;  
fintantoché prodMancanti > 0;  
potenza = potParziale;  
scrivi potenza;  
fine;
```

ISTRUZIONI IN
INGLESE

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
do  
potParziale = potParziale * base;  
prodMancanti = prodMancanti - 1;  
while prodMancanti > 0;  
potenza = potParziale;  
scrivi potenza;  
fine;
```

WHILE (...)

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
do  
  potParziale = potParziale * base;  
  prodMancanti = prodMancanti - 1;  
while (prodMancanti > 0);  
potenza = potParziale;  
scrivi potenza;  
fine;
```

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
do  
potParziale = potParziale * base;  
prodMancanti = prodMancanti - 1;  
while (prodMancanti > 0);  
potenza = potParziale;  
scrivi potenza;  
fine;
```

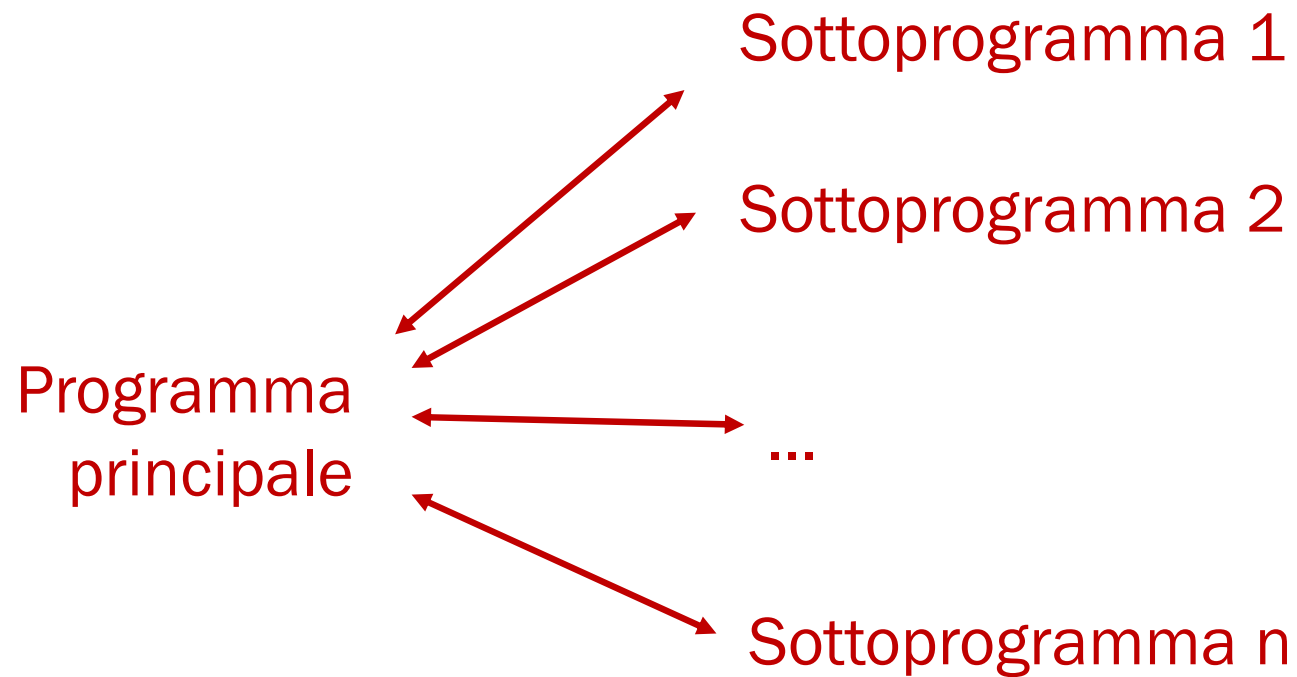
{...} RAGGRUPPA LE
ISTRUZIONI

```
leggi base;  
leggi esponente;  
potParziale = 1;  
prodMancanti = esponente;  
do  
{ potParziale = potParziale * base;  
  prodMancanti = prodMancanti - 1;  
} while (prodMancanti > 0);  
potenza = potParziale;  
scrivi potenza;  
fine;
```

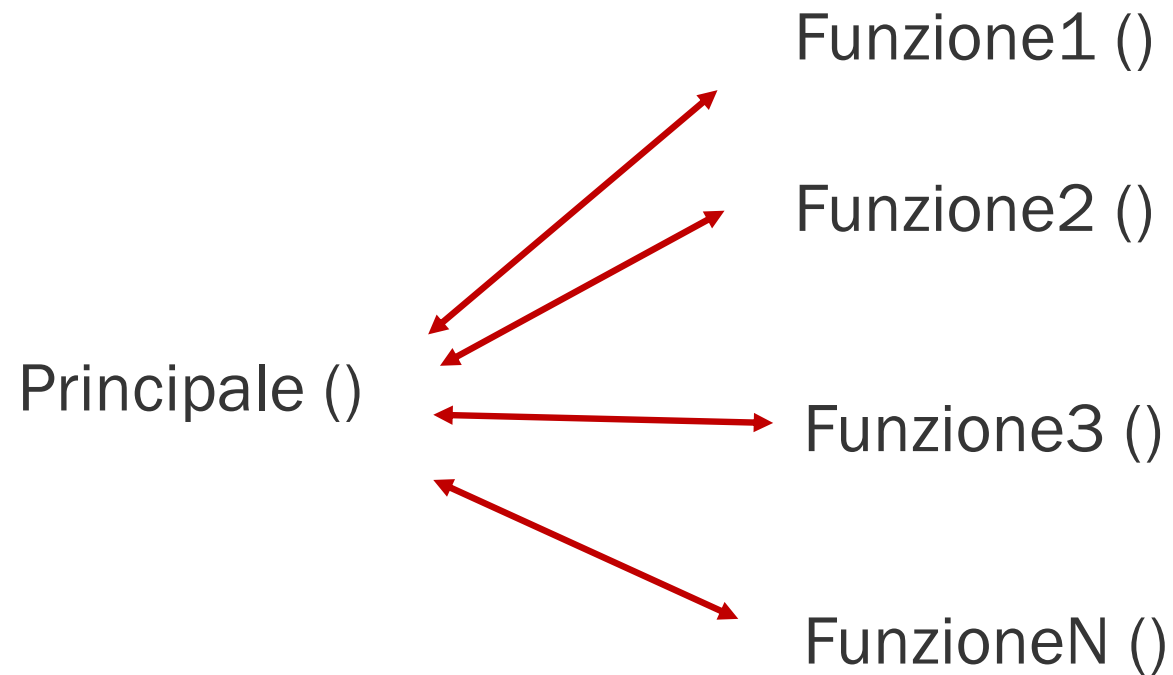
PROBLEMI E
SOTTOPROBLEMI



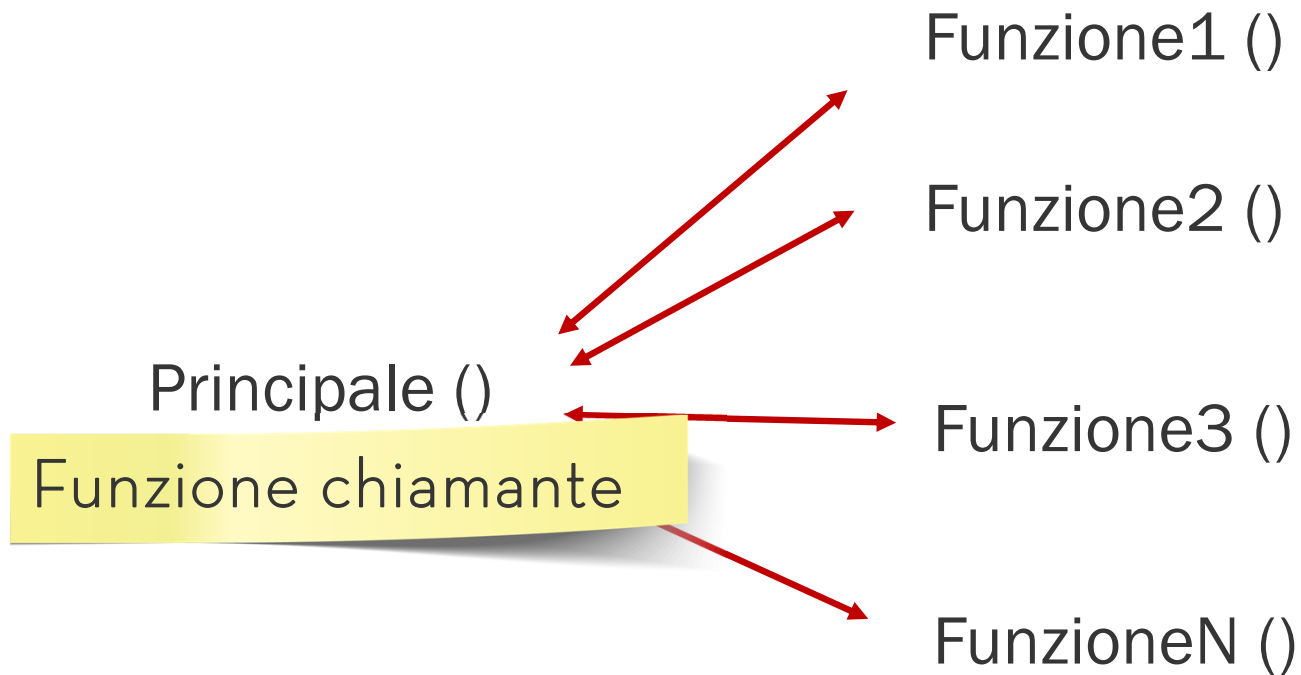
PROBLEMI E
SOTTOPROBLEMI



FUNZIONI



FUNZIONI



```
void main()  
{ leggi base;  
  leggi esponente;  
  potParziale = 1;  
  prodMancanti = esponente;  
  do  
  { potParziale = potParziale * base;  
    prodMancanti = prodMancanti - 1;  
  } while (prodMancanti > 0);  
  potenza = potParziale;  
  scrivi potenza;  
}
```