

## Quadrato di un numero. Analisi del metodo

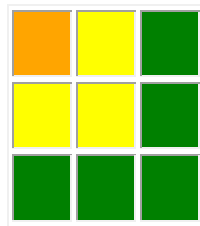
Si supponga che la mattonella in figura sia un quadrato di lato 1.



Se si vuole coprire un pavimento quadrato, avente il lato di lunghezza 2, si devono aggiungere altre 3 mattonelle disponendole come quelle gialle in figura:



Continuando ad incrementare la lunghezza del lato del quadrato, per coprire un pavimento quadrato di lato 3 bisogna aggiungere 5 mattonelle, disponendole come quelle verdi in figura:



Da questo esempio si può osservare che il quadrato di un numero  $N$  si può ottenere dalla somma dei primi  $N$  numeri dispari.

Ad esempio il quadrato  $Q$  del **numero 4** è ottenuto dalla somma dei primi 4 numeri dispari:  $1 + 3 + 5 + 7$ . Affinché il programma funzioni anche nel caso  $N=0$ , nella somma si consideri che il primo addendo sia 0, così che il quadrato di 4 si può anche scrivere:  $Q = 0 + 1 + 3 + 5 + 7$

---

### Algoritmo

Il procedimento di calcolo consiste nell'acquistare in una variabile  $N$  il valore del numero da elevare al quadrato. Poi si genera una sequenza dei primi  $N$  numeri *dispari* in una variabile  $D$ . Ogni numero dispari viene sommato alla variabile  $Q$ , che inizialmente contiene 0.

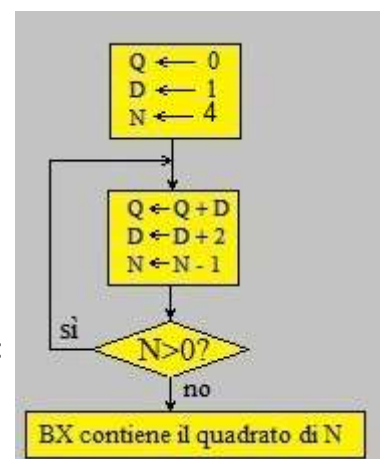
Le tre variabili individuate vengono memorizzate nei registri interni della CPU. Si decidono le seguenti posizioni:

la variabile ...	viene memorizzata nel registro ...
Q	<b>BX</b>
N	<b>CX</b>
D	<b>AX</b>

I valori iniziali delle variabili, supponendo di voler calcolare il quadrato di  $N = 4$  sono:

(variabile Q) **BX**  $\leftarrow$  0  
(variabile D) **AX**  $\leftarrow$  1  
(variabile N) **CX**  $\leftarrow$  4

Con questi valori iniziali si procede al calcolo del quadrato col seguente **flusso**:



Operazione	Simboli	valori in memoria
Si somma a Q il primo numero <i>dispari</i> D e si memorizza il risultato in Q:	$Q \leftarrow Q + D$	(Q) BX: 1
Si calcola il prossimo numero dispari:	$D \leftarrow D + 2$	(D) AX: 3
Si contano i dispari che restano da sommare:	$N \leftarrow N - 1$	(N) CX: 3
Se la quantità di dispari rimasti (N) è 0 allora il procedimento termina	$N > 0?$ (Si)	flag Z: falso
Si somma a Q il numero dispari contenuto in D	$Q \leftarrow Q + D$	(Q) BX: 4
Si calcola il prossimo numero dispari:	$D \leftarrow D + 2$	(D) AX: 5
Si conta quanti dispari restano da sommare:	$N \leftarrow N - 1$	(N) CX: 2
Se la quantità di dispari rimasti (N) è 0 il procedimento termina	$N > 0?$ (Si)	flag Z: falso
Si somma a Q il numero dispari contenuto in D	$Q \leftarrow Q + D$	(Q) BX: 9
Si calcola il prossimo numero dispari:	$D \leftarrow D + 2$	(D) AX: 7
Si conta quanti dispari restano da sommare:	$N \leftarrow N - 1$	(N) CX: 1
Se la quantità di dispari rimasti (N) è 0 il procedimento termina	$N > 0?$ (Si)	flag Z: falso
Si somma a Q il numero dispari contenuto in D	$Q \leftarrow Q + D$	(Q) BX: 16
Si calcola il prossimo numero dispari:	$D \leftarrow D + 2$	(D) AX: 9
Si conta quanti dispari restano da sommare:	$N \leftarrow N - 1$	(N) CX: 0
Se la quantità di dispari rimasti (N) è 0 il procedimento termina	$N > 0?$ (Si)	flag Z: vero
Sono stati sommati 4 dispari. Il procedimento è terminato. Il quadrato di N si trova in BX (Q)		

### Codifica in Assembly: programma *Quadrato.asm* in ambiente Jasmin

```

Q: dd 0           ; nella cella con indirizzo Q si memorizzerà il quadrato di N
N: dd 4           ; calcolare il quadrato di N
inizio:
  MOV BX, [Q]     ; Porta il contenuto di Q in BX
  MOV DX, [N]     ; Porta il contenuto di N in DX
  MOV AX, 1       ; Primo numero dispari
  MOV CX, DX      ; Il registro CX è un contatore
ripeti:
  ADD BX, AX      ; somma i numeri dispari
  ADD AX, 0002    ; passa al successivo numero dispari (somma 2)
  DEC CX          ; incrementa il contatore
  JCXZ fine       ; se il conteggio è giunto a 0 il procedimento termina (se CX è 0)
  JMP ripeti      ; altrimenti si ripete
fine:
  mov ax, 4c00h   ; uscita dal programma ... simulazione
  int 21h         ; in simulazione non si restituisce il controllo al sistema operativo

```

## Radice quadrata

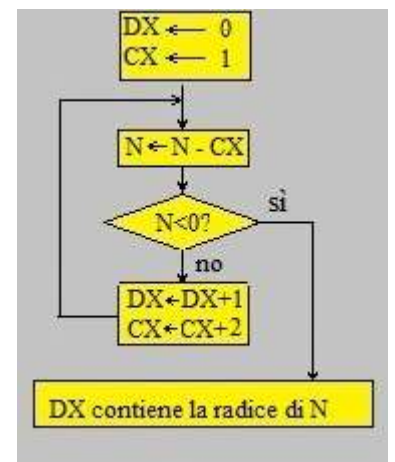
Per calcolare la radice quadrata di un numero  $N$  si devono **contare** quanti numeri *dispari* sono contenuti nel numero  $N$

Per ipotesi il **numero** di cui si vuole calcolare la radice quadrata è stato memorizzato nel registro AX.

Al termine del programma il risultato sarà disponibile nel registro  $DX$  ed eventualmente in una cella di memoria

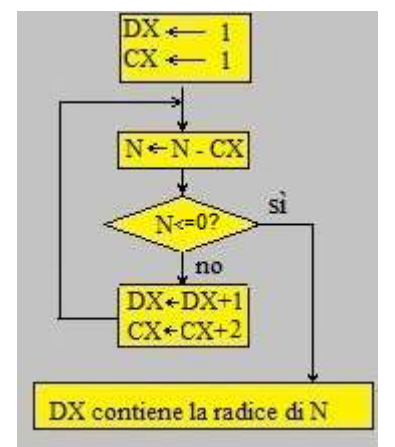
**Codifica in Assembly:** programma *Radice.asm* in ambiente Jasmin

```
R:dd 0 ;volendo il risultato anche in memoria
N:dd 16
inizio:
MOV AX, [N]
MOV DX, 0 ; contatore DX inizializzato a zero
MOV CX, 1 ; in CX i numeri dispari progressivi
ripeti:
SUB AX, CX
JL fine ;anche JB per numeri senza segno
INC DX ;al termine in DX il risultato
ADD CX, 2h
JMP ripeti
fine:
MOV [R],DX ;salva in memoria il risultato cioè DX
MOV AX, 4C00h
int 21h ;uscita dal programma (simulazione)
```



Equivalentemente, inizializzando il contatore ad uno:

```
R:dd 0
N:dd 81 ;numero di cui calcolare la radice
inizio:
MOV AX, [N]
MOV DX, 1 ; contatore DX inizializzato a uno
MOV CX, 1
ripeti:
SUB AX, CX
JLE fine
INC DX
ADD CX, 2h
JMP ripeti
fine:
MOV [R],DX ;salva in memoria il risultato cioè DX
MOV AX, 4C00h
int 21h
```



Tutte le istruzioni supportate da Jasmin:

<http://wwwi10.lrr.in.tum.de/~jasmin/commands.html>

[http://wwwi10.lrr.in.tum.de/~jasmin/tutorials\\_basic.html](http://wwwi10.lrr.in.tum.de/~jasmin/tutorials_basic.html) **tutorial**