Rappresentazione degli algoritmi
Notazione Lineare Strutturata

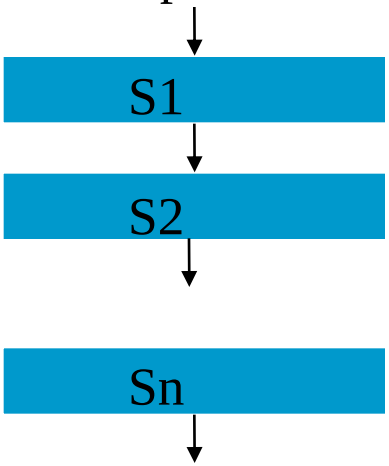
NOTAZIONE LINEARE STRUTTURATA (nls)

Si basa su un insieme di schemi di composizione che sono detti anche **costrutti di controllo strutturati**.

Ciascuno di essi verrà descritto sintatticamente da alcune regole che sono sostanzialmente le stesse che si ritrovano ne i linguaggi di programmazione tipo PASCAL o C

COSTRUTTO SEQUENZA (o begin end)

Questo schema prende ordinatamente un numero arbitrario di blocchi strutturati <S1>, <S2> … <Sn> e ne costruisce la sequenza secondo quanto espresso dal diagrammi a blocchi

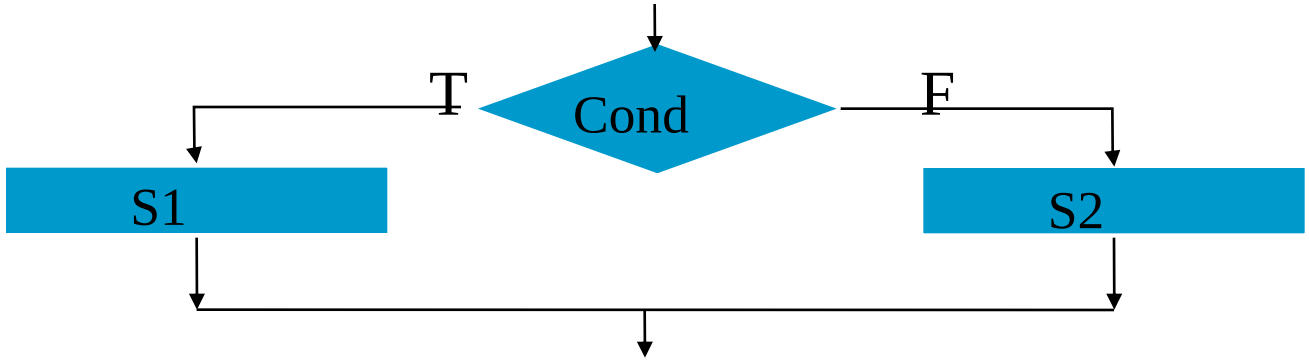


Sintassi NLS

{ S1;
S2;
.
.
.
Sn;
}

COSTRUTTO SCELTA (o if then else)

Questo schema valuta inizialmente una condizione <COND>, se questa e VERA, viene successivamente eseguito in blocco strutturato <S1> altrimenti viene eseguito il blocco strutturati <S2> secondo quanto espresso dal diagrammi a blocchi

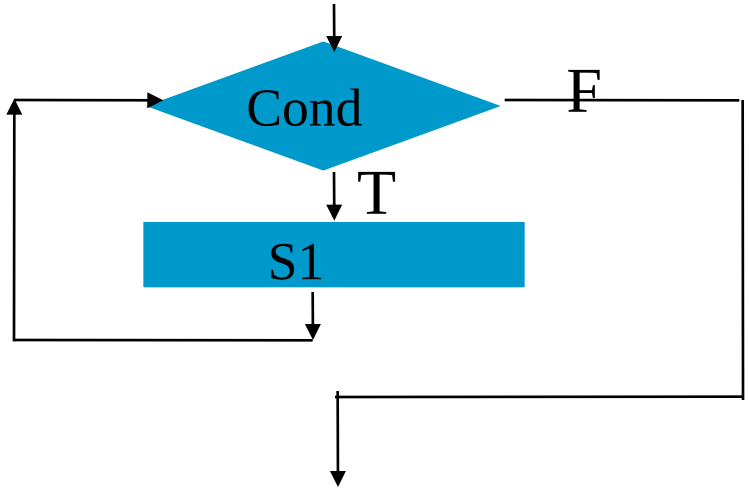


Sintassi NLS

if (<COND>) S1 else S2;

COSTRUTTO CICLO (o while do)

Questo schema prevede una condizione <COND>, detta **GUARDIA** del ciclo, da valutare inizialmente. Se questa è falsa si esce immediatamente dal costrutto, altrimenti si esegua ciclicamente il blocco strutturato <S>, detto **CORPO** del ciclo fino a quando la condizione resterà VERA



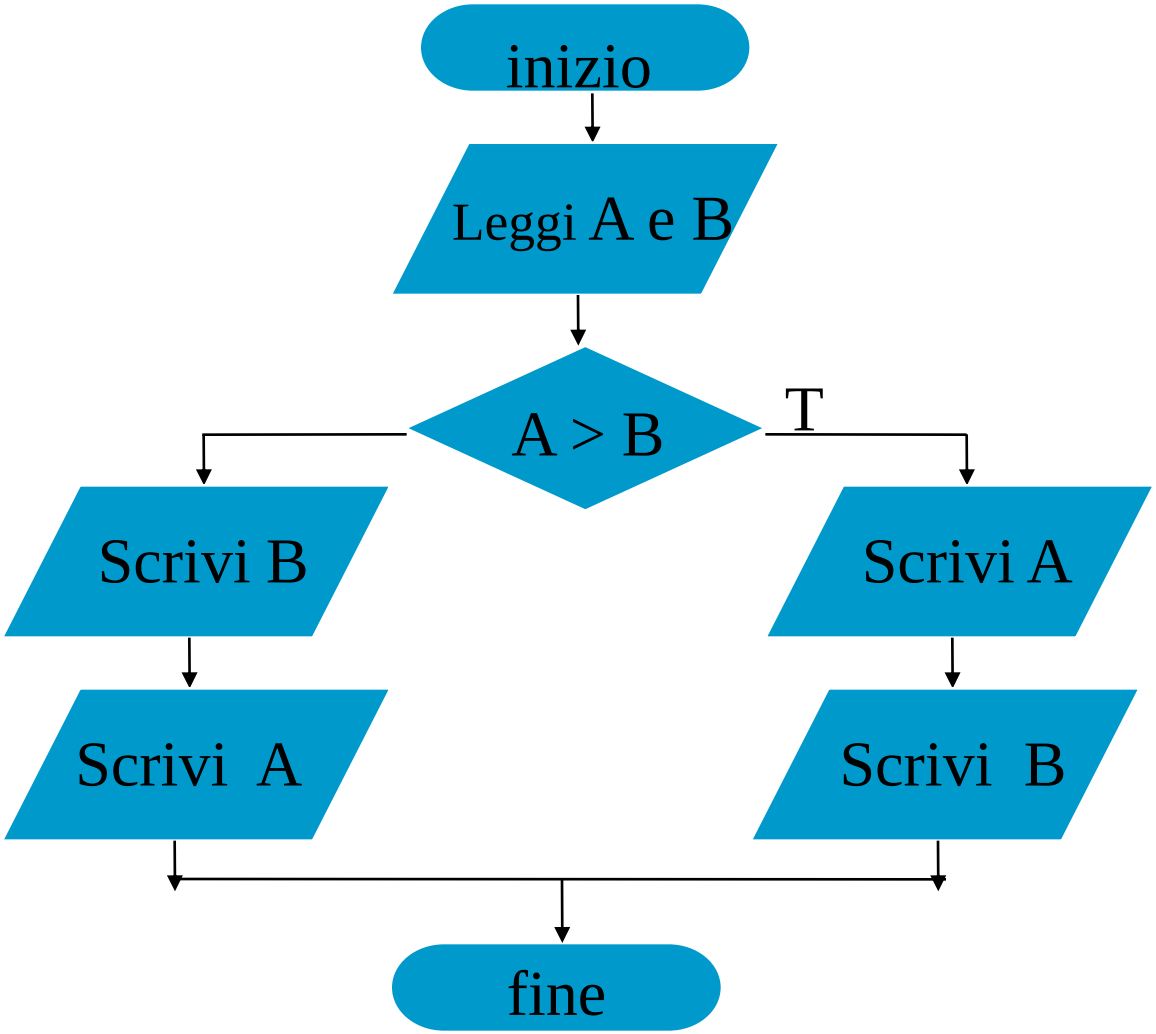
Sintassi NLS

while (<COND>) S;

Teorema di Böhm-Jacopini

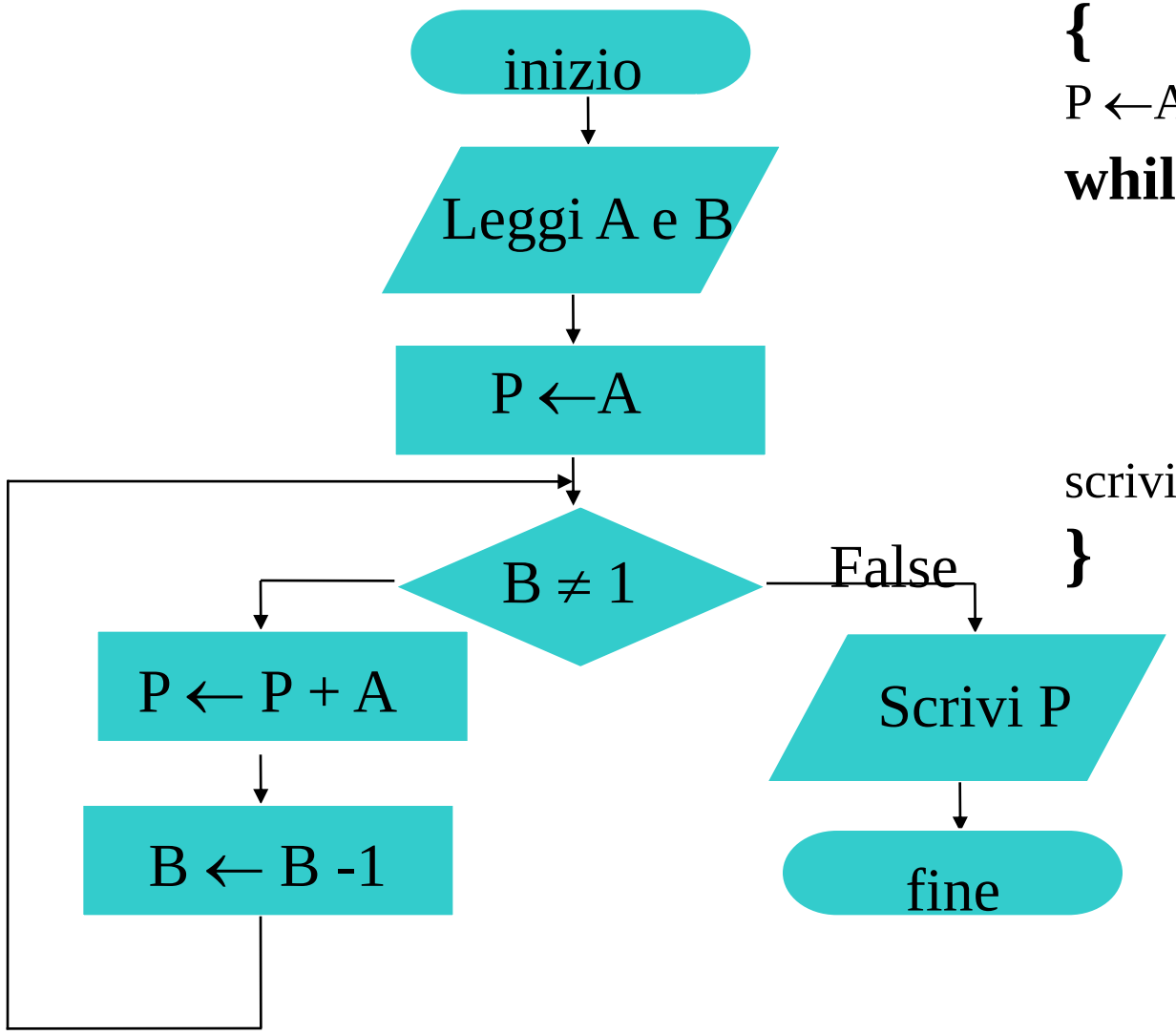
Qualunque algoritmo può essere implementato utilizzando tre sole strutture, la **sequenza**, la **selezione** ed il **ciclo**, da applicare ricorsivamente alla composizione di istruzioni elementari (1966)

Dato due numeri interi stampare prima il più grande e poi il più piccolo



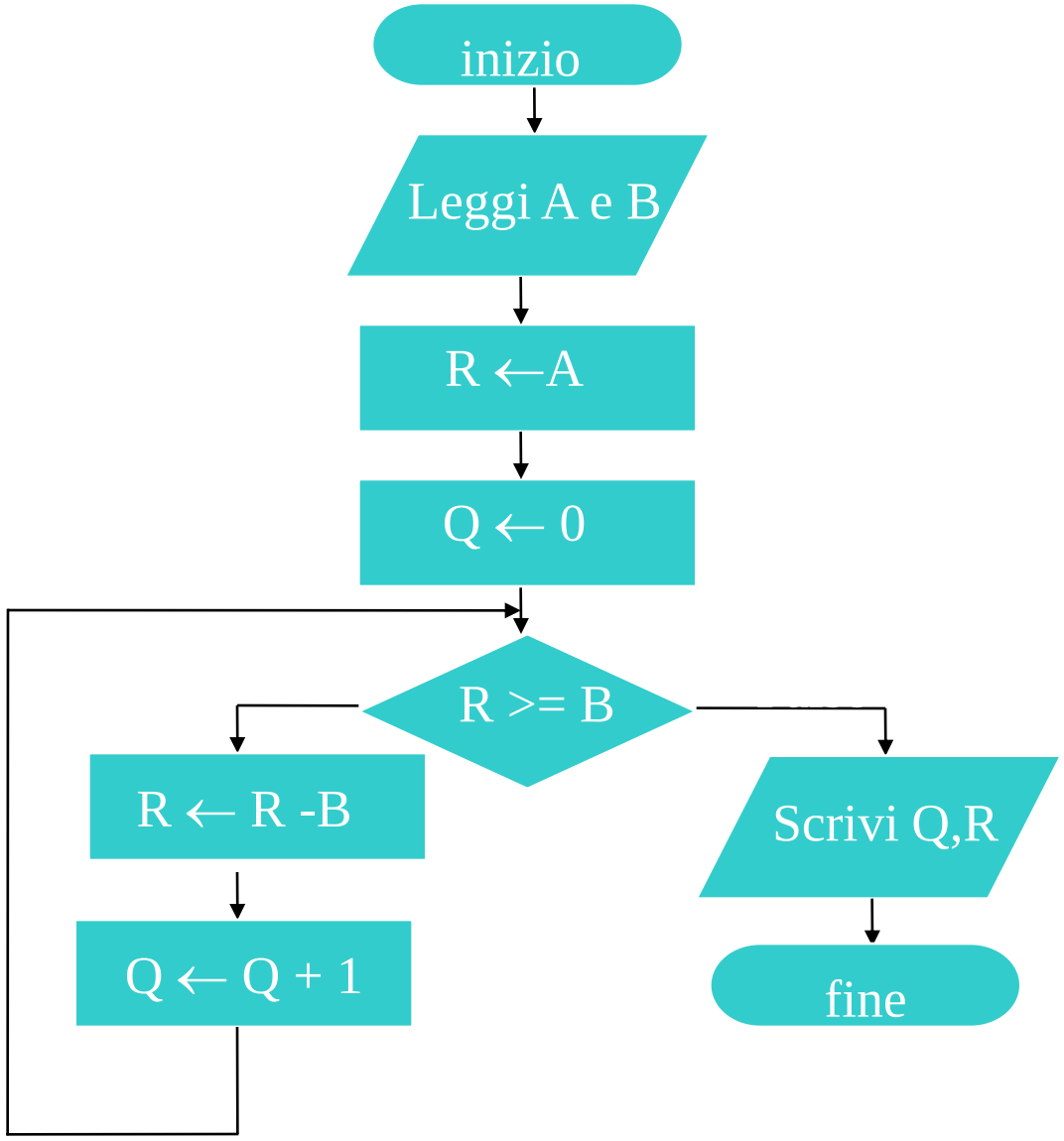

```
{
leggi A e B;
 if (A>B ) {
        scrivi A;
        scrivi B;
    }
   else {
        scrivi B;
       scrivi A:
   };
}
```

Dato due numeri interi positivi effettuare il loro prodotto con il metodo delle addizioni successive



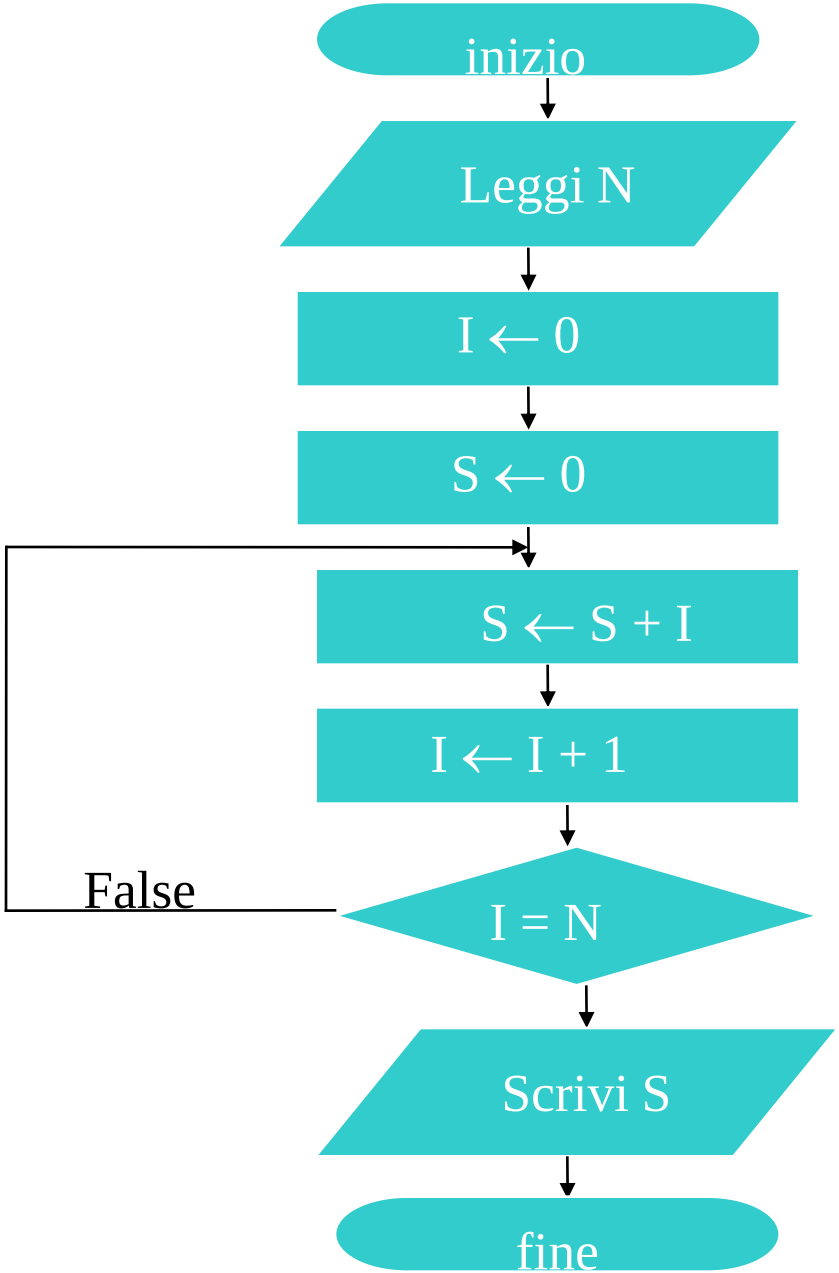

```
{          leggi A e B;
P ←A;
while (B ≠ 1)
          { P ← P + A;
            B ← B -1;
          }
scrivi P;
}
```

Dato due numeri interi positivi diversi da zero calcolare quoziente e resto con il metodo delle sottrazioni successive




```
{          leggi A e B;
R ←A;
Q ←A;
while (R >=B)
          { R ← R - B;
            Q ← Q +1;
          };
scrivi Q e R;
}
```

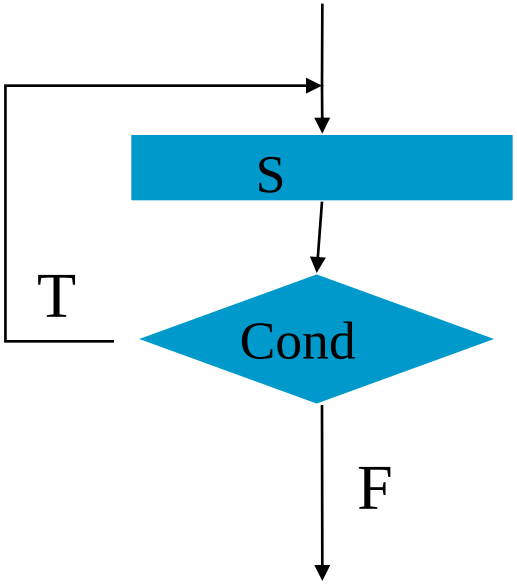
Sommare i primi N interi positivi




```
{
leggi N;
I ←0;
S ←0;
while (I < N)
        {
        S ← S + I;
        I ← I +1;
        };
scrivi S;
}
```

COSTRUTTO CICLO (o do while)

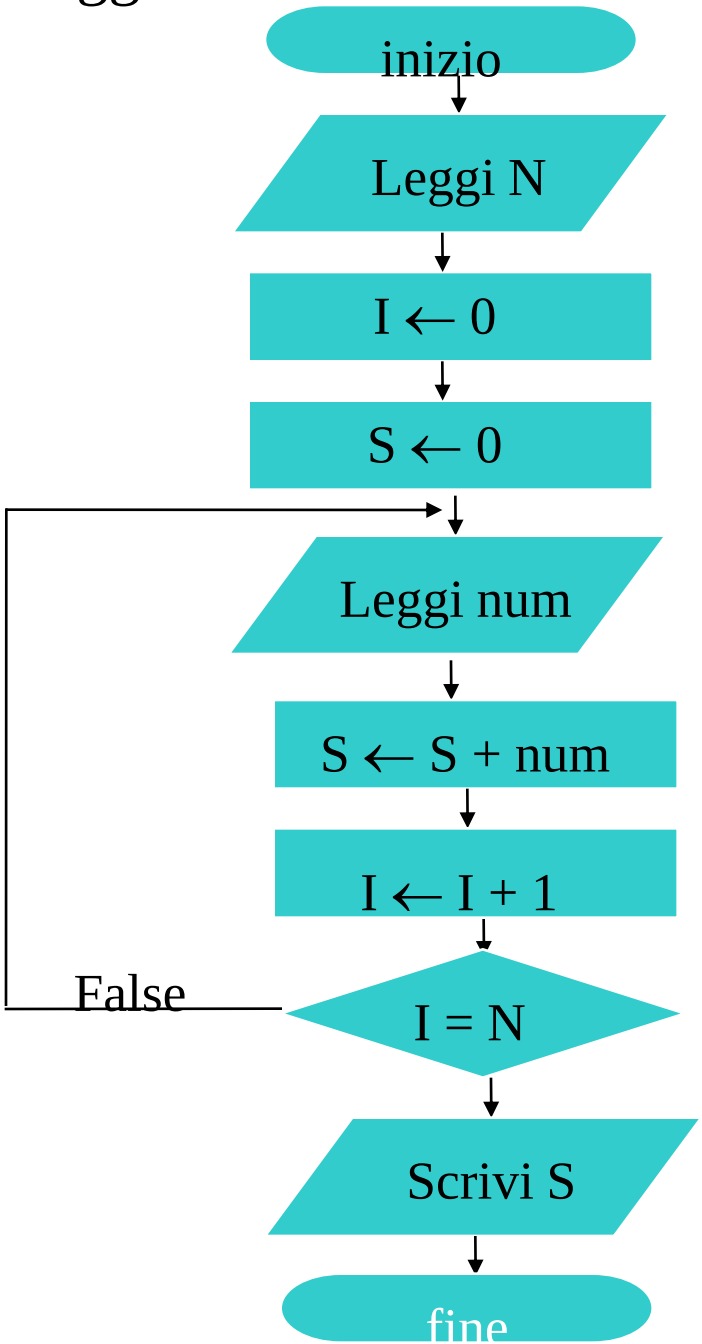
Questo schema prevede una condizione <COND> da valutare alla fine di ogni ciclo. Se questa è falsa non si esegue il ciclo successivo e si esce immediatamente dal costrutto, altrimenti si esegua ciclicamente il blocco strutturato <S>, detto **CORPO** del ciclo fino a quando la condizione resterà VERA



Sintassi NLS

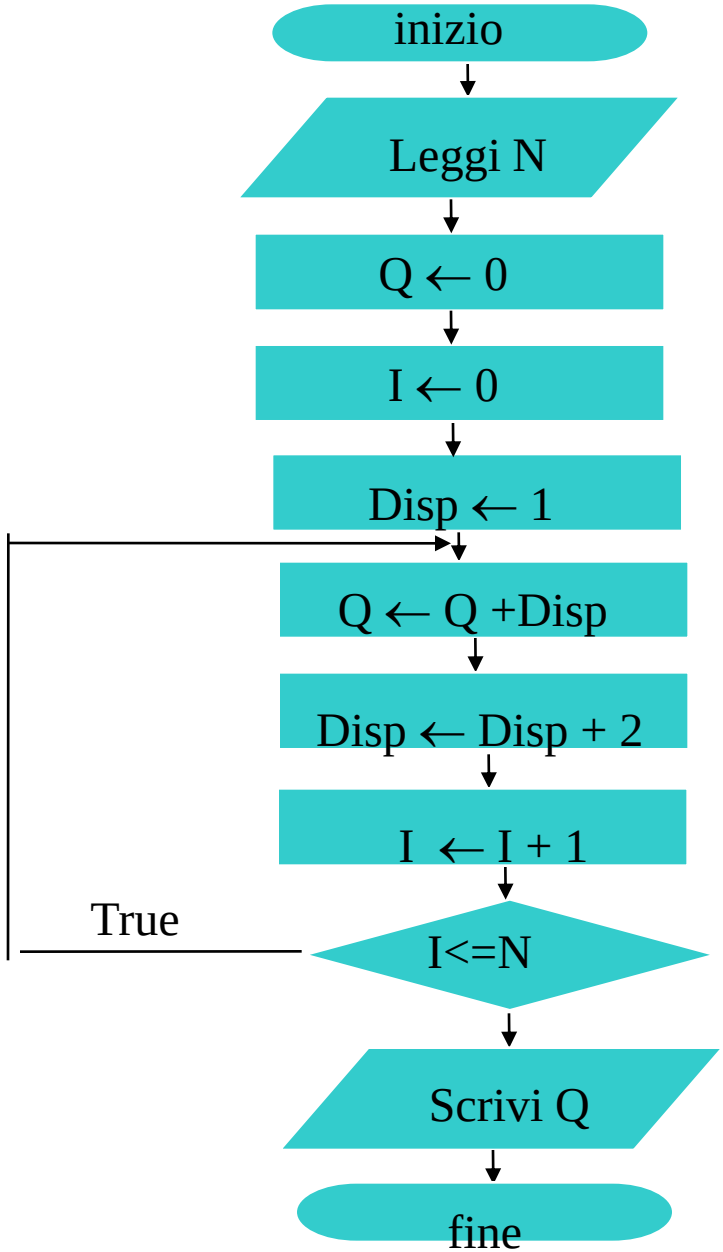
do S while (<COND>) ;

Leggere N numeri e scrivere la loro somma.



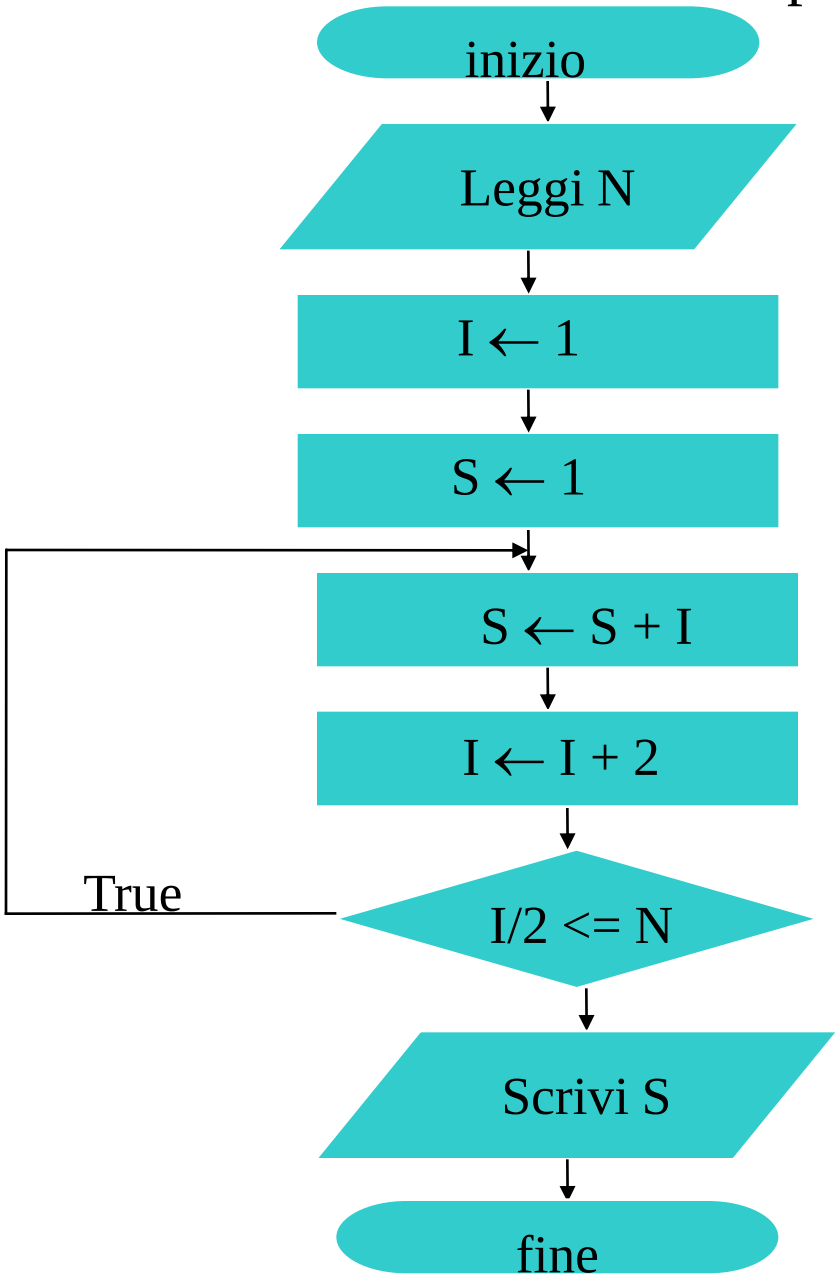

```
{           leggi N;
I ←0;
S ←0;
do
            { leggi Num;
              S ← S + Num;
              I ← I +1;
            }
while (I ≠ N);
scrivi S;
}
```

Calcolare il quadrato di un numero N intero positivo utilizzando la somma dei primi N numeri dispari




```
{        leggi N;
Q ← 0;
I ←0;
Disp ← 1;
 do
          { Q ← Q + Disp;
            Disp ← Disp + 2;
            I ← I +1;
          }
while (I<=N);
scrivi Q
}
```

Sommare i primi N numeri dispari




```
{
leggi N;
I ←1;
S ←1;
do
        { S ← S + I;
          I ← I +2;
        }
while (I/2<=N);
scrivi S;
}
```