



Informatica per l'Ingegneria

Corsi M – N

A.A. 2023/2024

Angelo Cardellicchio

11 – Pseudocodifica



La pseudocodifica

- Per **pseudocodifica** si intende un linguaggio per la descrizione di algoritmi secondo le regole della programmazione strutturata.
- La descrizione di un algoritmo in pseudocodifica si compone di due parti:
 - la **dichiarazione delle variabili** usate nell'algoritmo...
 - ...e la **descrizione delle azioni** dell'algoritmo.



Tipo delle variabili (1)

- Il **tipo** di una variabile indica l'insieme dei valori che possono essere ad essa assegnati.
- Su costanti e variabili di un tipo è possibile effettuare le operazioni che sono proprie di quel tipo, oltre che tutte le operazioni di confronto.
- Sono permessi quattro tipi, ovvero **integer**, **real**, **boolean** e **string-q**.



Tipo delle variabili (2)

- **Integer:** sono le variabili cui possono essere assegnati numeri interi; le costanti di tipo integer sono numeri interi, ad esempio, 1, -3, 150.
- **Real:** sono le variabili cui possono essere assegnati numeri razionali; le costanti real possono essere rappresentate in notazione decimale, con un punto che separa la parte intera dalla parte decimale (ad esempio, 5.16, 12.367, -123., 0.005) o in notazione scientifica ($23.476 \times 10^3 = 23476$, $456.985 \times 10^{-3} = 0.456095$).



Tipo delle variabili (3)

- **Boolean**: sono le variabili cui possono essere assegnati i valori logici. Le costanti logici sono **true** e **false**.
- **String-q**: sono le variabili cui possono essere assegnate parole (o **stringhe**) costituite da q caratteri. Le costanti string-q sono costituite da parole di q caratteri racchiusi tra apici (che non fanno parte della costante). Ad esempio, *'FABIO'* è una costante string-5, *' + '* è una costante string-1, mentre *'124'* è string-3.



Dichiarazione delle variabili

- La dichiarazione delle variabili nella pseudocodifica è un elenco, preceduto dalla parola **var**, delle variabili sulle quali l'algoritmo opera.
- Le variabili sono suddivise per tipo:
 - quelle dello stesso tipo sono separate l'una dall'altra da una virgola;
 - l'elenco delle variabili dello stesso tipo è seguito da due punti ed indicazione del tipo;
 - gli elenchi di variabili di tipo diverso sono separati dal punto e virgola;
 - l'ultimo elenco è seguito da un punto.
- Ad esempio:

```
var i, j, a(20): integer;  
p, q: real;  
nome: string-20;  
sw: boolean.
```



Descrizione delle azioni (1)

- Gli schemi di flusso fondamentali sono descritti utilizzando convenzioni linguistiche: ad ogni schema strutturato corrisponde una convenzione linguistica.
- La descrizione di un algoritmo deve soddisfare le seguenti regole:
 - La prima azione dell'algoritmo è preceduta dalla parola **begin**.
 - L'ultima azione dell'algoritmo è seguita dalla parola **end**.
 - L'azione di lettura è rappresentata dalla parola **read**.
 - L'azione di scrittura è rappresentata dalla parola **write**.
 - Lo schema di sequenza di n flussi S_1, S_2, \dots, S_n è rappresentato come:

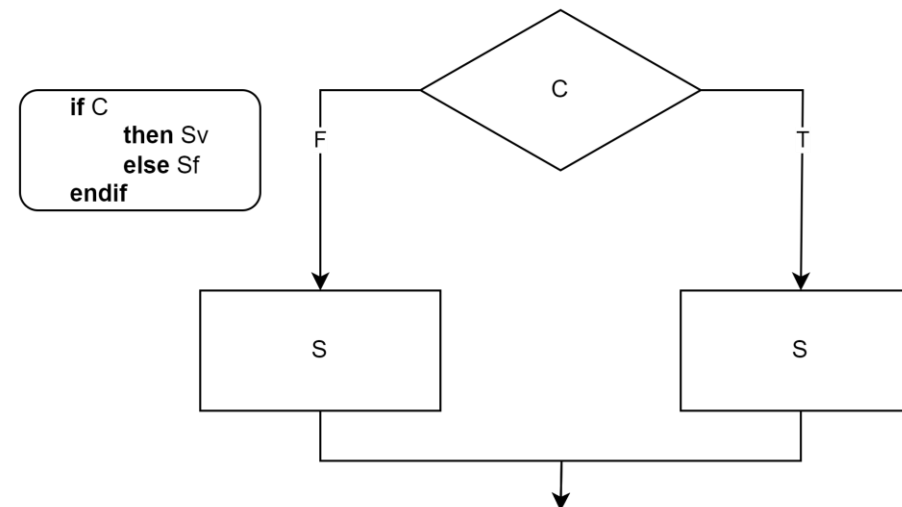
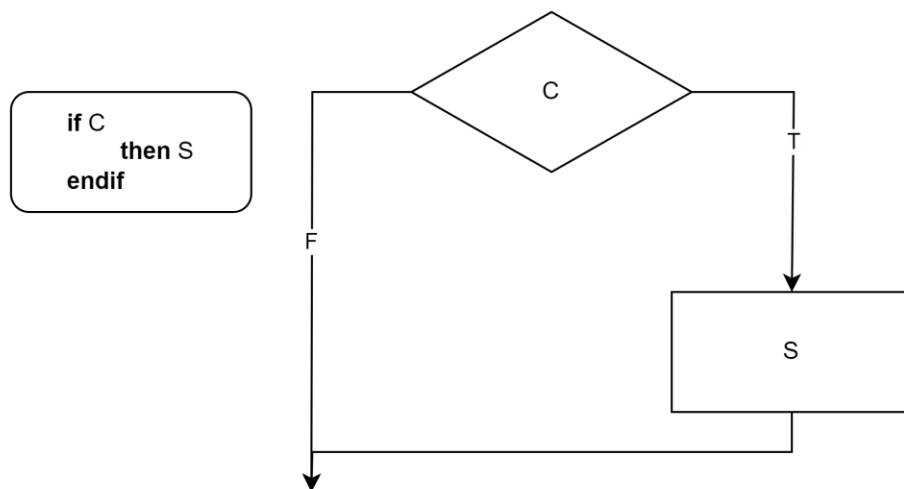
$S_1;$

$S_2;$

$\dots S_n;$

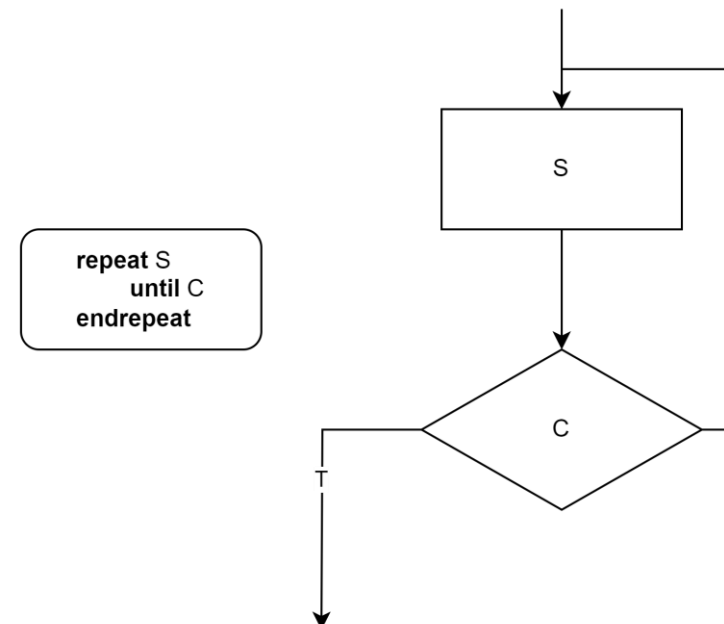
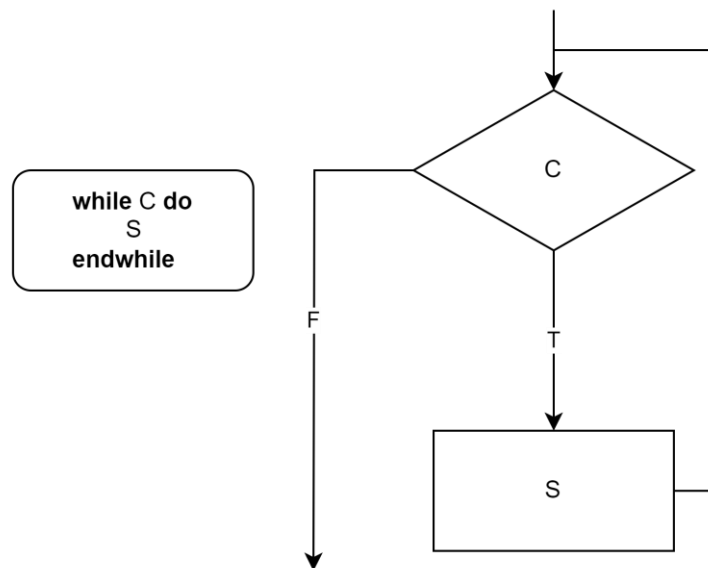
Descrizione delle azioni (2)

- Gli schemi di selezione sono rappresentati come:



Descrizione delle azioni (3)

- Gli schemi di iterazione sono rappresentati come:



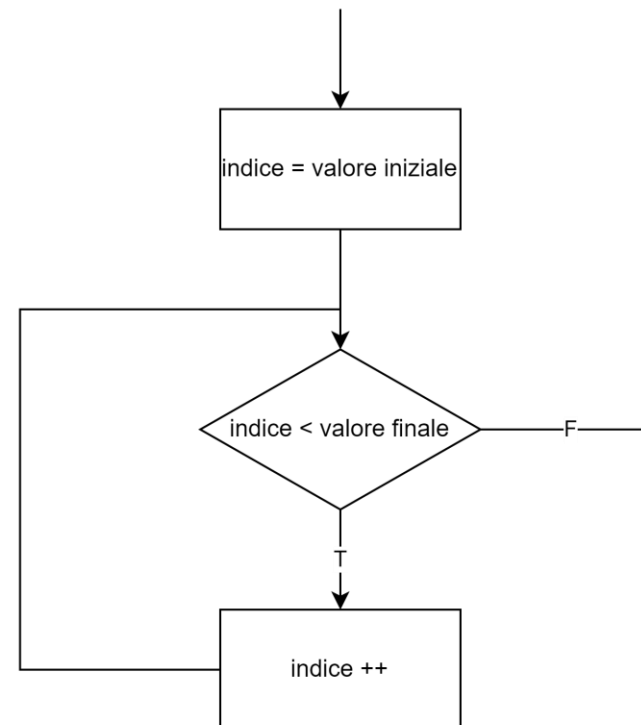


Descrizione delle azioni (4)

- Esistono convenzioni linguistiche alternative in relazione a particolari schemi di flusso.

- Esempio:** Ciclo enumerativo

- Se il valore di incremento è unitario, la parte di **step incremento** della frase **for...endfor** può essere omessa.*



for indice from valore iniziale to valore finale step incremento do
S
endfor



Esempi (1)

- **Esempio:** Algoritmo per il calcolo del vettore somma di due vettori di numeri razionali.

```
var      a(100), b(100), c(100) : real;  
         i, n : integer.  
begin  
    read n;  
    for i from 1 to n step 1 do  
        read a(i), b(i);  
        c(i) ← a(i) + b(i);  
        write c(i)  
    endfor  
end
```



Esempi (2)

- **Esempio:** Algoritmo per il calcolo del massimo elemento di un vettore di numeri razionali.

```
var    max, v(100) : real;  
        i, n : integer.  
begin  
    read n;  
    for i from 1 to n step 1 do  
        read v(i);  
    endfor  
    max ← v(1);  
    for i from 2 to n step 1 do  
        if max < v(i)  
            then max ← v(i)  
        endif  
    endfor  
end
```



Esempi (3)

- **Esempio:** Algoritmo per il calcolo delle radici di un'equazione di secondo grado.

```
var      x1, x2, a, b, c, delta : real.

begin
    read a, b, c;
    delta ←  $b^2 - 4ac$ ;
    if delta < 0
        then write 'non esistono radici reali'
        else if delta = 0
            then      x1 ←  $-b/2^{\circ}$ ;
                     x2 ← x1
            else      x1 ←  $(-b + \text{sqrt}(\text{delta}))/2a$ ;
                     x2 ←  $(-b - \text{sqrt}(\text{delta}))/2a$ ;
        endif
        write x1, x2
    endif
end
```



Esempi (4)

- **Esempio:** Algoritmo per il calcolo della somma di una sequenza di numeri.

```
var    a, s: real;
      n: integer.
begin
    read n;
    s ← 0;
    repeat
        read a;
        s ← s + a;
        n ← n - 1;
    until n = 0
    endrepeat
    write s
end
```



Domande?

42