

06. La programmazione strutturata

Corso di Algoritmi e Linguaggi di Programmazione Python/C

Outline

- GOTO e Spaghetti Code
- Il Teorema di Böhm – Jacopini
- Le Strutture di Controllo
 - Sequenza
 - Selezione
 - I costrutti IF – THEN – ELSE e SWITCH
 - Iterazione
 - I costrutti FOR e WHILE
- Note

GOTO e Spaghetti Code

- Anni '60: **Spaghetti Code!**
 - *Gli algoritmi allora si basavano prevalentemente sul costrutto **go to**, che indicava al programma l'istruzione verso la quale 'saltare'.*
 - *Vedete un esempio di queste meraviglie a lato.*
- Approccio fortemente criticato
 - *Ad esempio, Dijkstra ne discusse gli effetti deleteri in **Go To Statement Considered Harmful***
- Codice strutturato più semplice da strutturare e mantenere!

```
10 int i = 0
20 i = i + 1
30 i = i + 2
40 if i <= 10 then goto 70
50 print "Programma terminato."
60 end
70 print i & " al quadrato = " & i * i
80 goto 20
```



```
for (int i = 0; i <= 10; i++)
{
    print(i & " al quadrato = " & i * i);
}
print("Programma terminato");
```

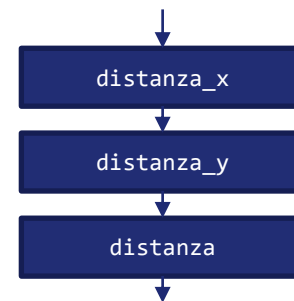
Il Teorema di Böhm - Jacopini

- Il Teorema di Böhm – Jacopini stabilisce che:
Ogni algoritmo può essere costruito a partire da tre strutture di controllo fondamentali, ovvero sequenza, selezione ed iterazione.
- Il teorema ha avuto un forte impatto nel passaggio dalla programmazione non strutturata a quella strutturata.
- Le sue implicazioni sono, chiaramente, estremamente importanti.

Le Strutture di Controllo - Sequenza

- Istruzioni realizzate sequenzialmente, ovvero l'una in cascata all'altra
- Es. calcolo distanza euclidea

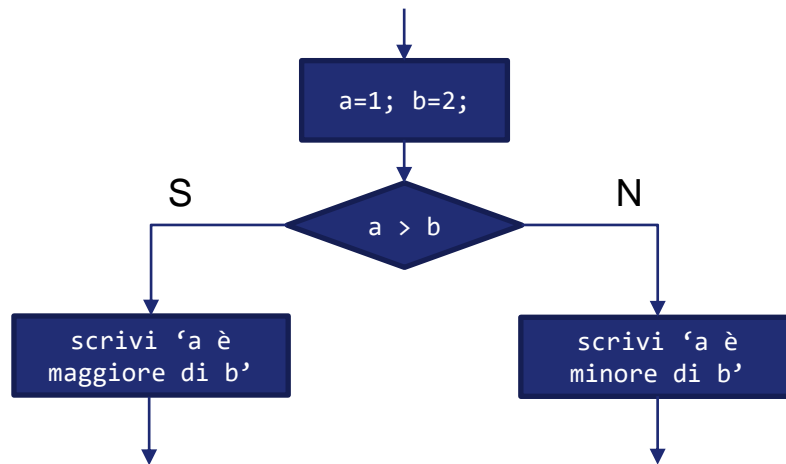
```
distanza_x = (x_a - x_b)^2  
distanza_y = (y_a - y_b)^2  
distanza = (distanza_x + distanza_y)^(1/2)
```



Le Strutture di Controllo - Selezione

- La struttura di *selezione* ci permette di scegliere tra due diverse opzioni in base ad una condizione.
- Per farlo, si usa il costrutto IF – THEN – ELSE
- I due rami del programma sono divergenti e mutualmente esclusivi

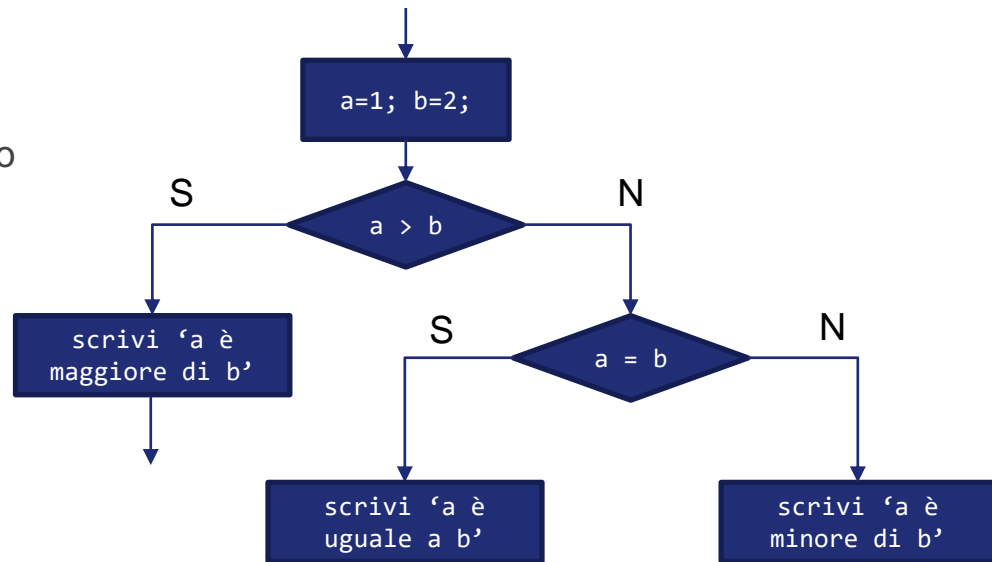
```
a = 1;  
b = 2;  
if (a > b):  
    then scrivi 'a è maggiore di b'  
else:  
    scrivi 'b è maggiore di a'
```



I costrutti IF – THEN – ELSE e SWITCH (1)

- Esistono due modi per implementare la selezione.
- Il primo è il costrutto **IF – THEN – ELSE**, visto nella slide precedente.
- Il secondo è il costrutto **SWITCH**.
- Il costrutto IF – THEN – ELSE può gestire diversi rami mediante l'ELSE IF.

```
a = 1;
b = 2;
if (a > b):
    then scrivi "a è maggiore di b"
else if (a uguale b):
    then scrivi "a è uguale a b"
else:
    scrivi "b è minore di a"
```



I costrutti IF – THEN – ELSE e SWITCH (2)

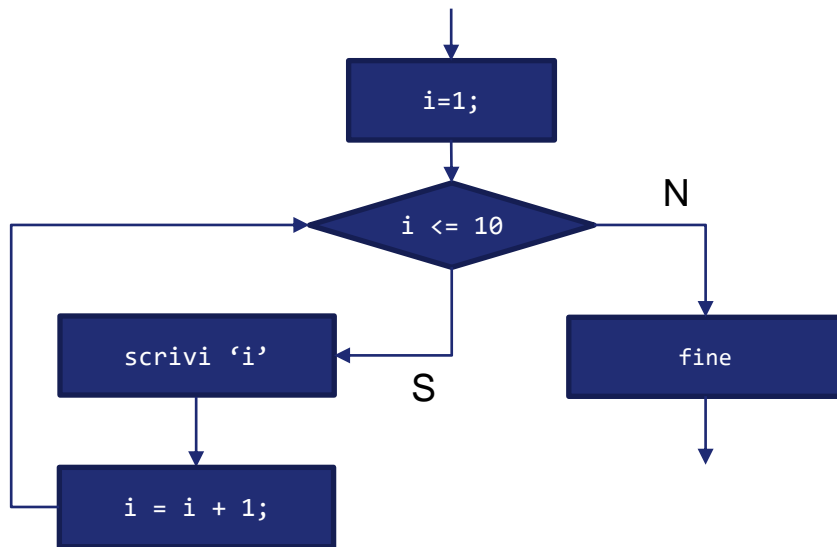
- Lo SWITCH è anch'esso usato per modellare diverse possibilità (rami).
- Verifica che una variabile assuma certi valori (**case**) ed un comportamento di **default**.
- Di solito, i case sono ben definiti, e non si riferiscono a dei range di possibili valori.

```
a = 1;
switch (a)
  case 1:
    scrivi "Uguale ad uno!";
  case 2:
    scrivi "Uguale a due!";
  case 3:
    scrivi "Uguale a tre!";
  default:
    scrivi "Non saprei!";
```


Le Strutture di Controllo - Iterazione

- E' una struttura di controllo che *reitera* (ovvero, *ripete*) un'istruzione fino al verificarsi di una condizione.
- Quando la condizione non è più verificata, il programma prosegue.

```
i=1;  
for (i che va da 1 a 10):  
    scrivi 'i';  
endfor  
scrivi 'fine';
```



I costrutti FOR e WHILE

- Esistono due modi per implementare l'iterazione.
- Il primo è il costrutto **FOR**, utilizzato per ripetere un'istruzione un certo numero di volte.
- Il secondo è il costrutto **while**, usato per ripetere un'istruzione fino a che è verificata una certa condizione.

```
i = 1;
for (i che va da 1 a 10):
    scrivi 'i';
endfor
scrivi 'fine';
```

```
condizione = vero;
while (condizione diverso da falso):
    do [...] // istruzioni
    aggiorna condizione;
endwhile
```

Note e considerazioni

- È importante porre particolare attenzione all'aggiornamento della condizione nel ciclo WHILE: se ciò non avviene, potrebbe essere impossibile uscirne!
- Lo SWITCH normalmente non prevede casi in cui si valutano dei range; è molto più semplice usare un IF – THEN – ELSE qualora sia questo il caso.
 - È comunque possibile implementare uno switch con dei case che riguardano dei range.
- A seconda del linguaggio di programmazione, lo SWITCH **potrebbe** essere più performante dell'IF – THEN – ELSE.

Esercizi

- *Scrivere un diagramma di flusso che confronti due numeri letti da un input esterno. Scrivere a schermo se i numeri sono uguali o meno.*
- *Scrivere un algoritmo che aumenti il valore di un numero letto in ingresso di dieci unità in maniera iterativa, e poi verifichi che il valore del numero stesso sia superiore a quindici.*
- *Scrivere un algoritmo che generi numeri interi casuali fino a che l'ultimo numero generato non sia superiore a 10. Si supponga che i numeri casuali siano generati mediante un'istruzione chiamata "genera magia". In particolare, scrivendo: `a = genera magia`; supponiamo che ad `a` sia assegnato un certo valore intero casuale.*

Domande?

42