

10. Programmazione Strutturata

Corso di Informatica

Outline

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

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 if i <= 10 then goto 60
40 print "Programma terminato.«
50 end
60 print i & " al quadrato = " & i * i
70 goto 20
```



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

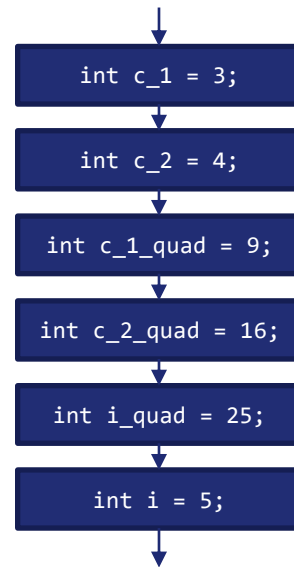
Il Teorema di Böhm - Jacopini

- Il Teorema di Böhm – Jacopini ha avuto un forte impatto nel passaggio verso la programmazione strutturata.
- Stabilisce che **ogni algoritmo può essere costruito a partire da tre strutture di controllo, ovvero sequenza, selezione ed iterazione.**
- Le implicazioni sono importantissime:
 - *possiamo ricondurre tutto a solamente tre strutture;*
 - *di conseguenza, non abbiamo bisogno del **goto**...*
 - *...e possiamo dire addio allo Spaghetti Code!*

Le Strutture di Controllo - Sequenza

- Abbiamo già visto il concetto di sequenza di istruzioni quando abbiamo parlato di diagrammi di flusso.

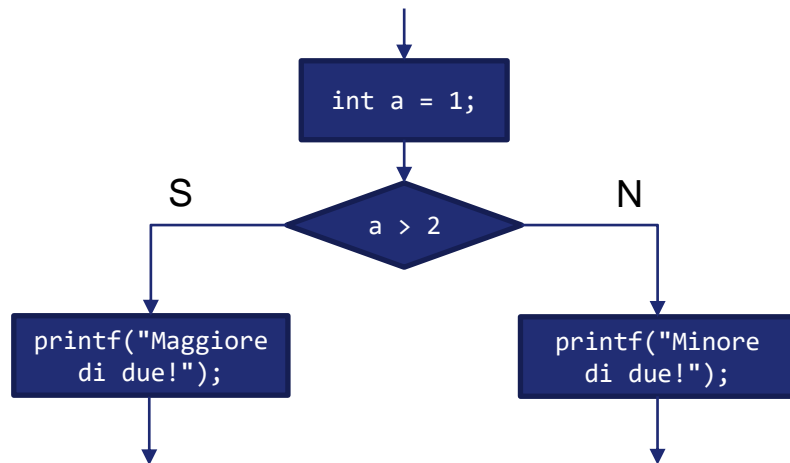
```
int c_1 = 3;  
int c_2 = 4;  
int c_1_quad = c_1 * c_1;  
int c_2_quad = c_2 * c_2;  
int i_quad = c_1_quad + c_2_quad;  
int i = sqrt(i_quad);
```



Le Strutture di Controllo - Selezione

- La struttura di *selezione* ci permette di scegliere tra due diverse opzioni in base ad una condizione.
- Comporta la divergenza di due **rami (branch)** del programma.

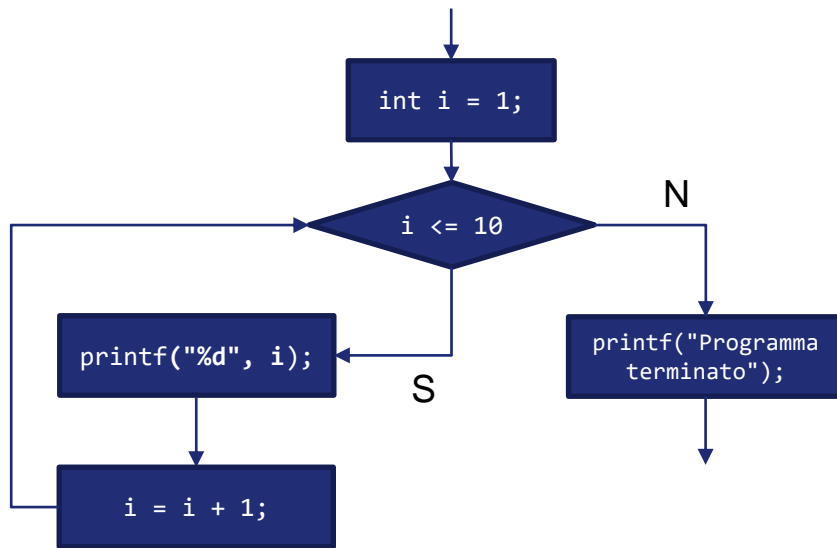
```
int a = 1;  
if (a > 2)  
{  
    printf("Maggiore di due!");  
} else {  
    printf("Minore di due!");  
}
```



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.

```
for (int i = 0; i <= 10; i++)  
{  
    print("%d", i);  
}  
print("Programma terminato");
```



I Costrutti IF / THEN / ELSE e SWITCH (1)

- Esistono due modi per implementare la selezione.
- Il primo è il costrutto *if / then / else*.

```
int a = 1;
if (a > 2)
{
    printf("Maggiore di due!");
} else {
    printf("Minore di due!");
}
```

- Il costrutto può gestire anche diverse possibilità (rami) usando l'istruzione *else if*.

```
if (a > 2)
{
    printf("Maggiore di due!");
} else if (a == 1) {
    printf("Uguale ad uno!");
} else {
    printf("Minore di due!");
}
```


I Costrutti IF / THEN / ELSE e SWITCH (2)

- Lo **switch** è anch'esso usato per modellare diverse possibilità (rami).
- Ha diversi casi (**case**) ed un comportamento di **default**.

```
int a = 1;
switch (a) {
    case 1:
        printf("Uguale ad uno!");
        break;
    case 2:
        printf("Uguale a due!");
        break;
    case 3:
        printf("Uguale a quattro!");
        break;
    default:
        printf("Non saprei!");
        break;
}
```

I Cicli FOR e WHILE

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

```
for (int i = 1; i <= 10; i++)  
{  
    print("Il quadrato di i è %d", i*i);  
}
```

```
i = 0;  
while (i <= 10)  
{  
    print("Il quadrato di i è %d", i*i);  
    i++;  
}
```

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, usando ad esempio delle istruzioni booleane.*
- A seconda del compilatore, lo **switch potrebbe** essere più performante dell'**if/then/else**.
 - *Questo è legato al fatto che sono utilizzate delle strutture dati ottimizzate chiamate **jump table**, che trattano il valore associato ad ogni case come un indice ad una serie di istruzioni.*

Domande?

42