

# Indentazione (1/2)

- Per rendere leggibile un programma si utilizza la pratica di *indentare* il codice

```
int main()  
{  
    int i ;  
    ...  
    if (...)  
        cout<<"messaggio" ;  
    return 0 ;  
}
```

1 carattere di tabulazione (TAB)

2 caratteri di tabulazione (TAB)

# Indentazione (2/2)

- Se C1 e' la colonna rispetto alla quale sono allineate
  - l'intestazione di una funzione,
  - una istruzione condizionale o una istruzione iterativa,
  - o l'inizio di una istruzione composta
- Tutte le istruzioni appartenenti al loro corpo, devono essere allineate a partire da una colonna C2,
- spostata a destra di un numero prefissato di spazi (o meglio, caratteri di tabulazione) rispetto a C1.

# Esercizio (1)

Scrivere un programma che:

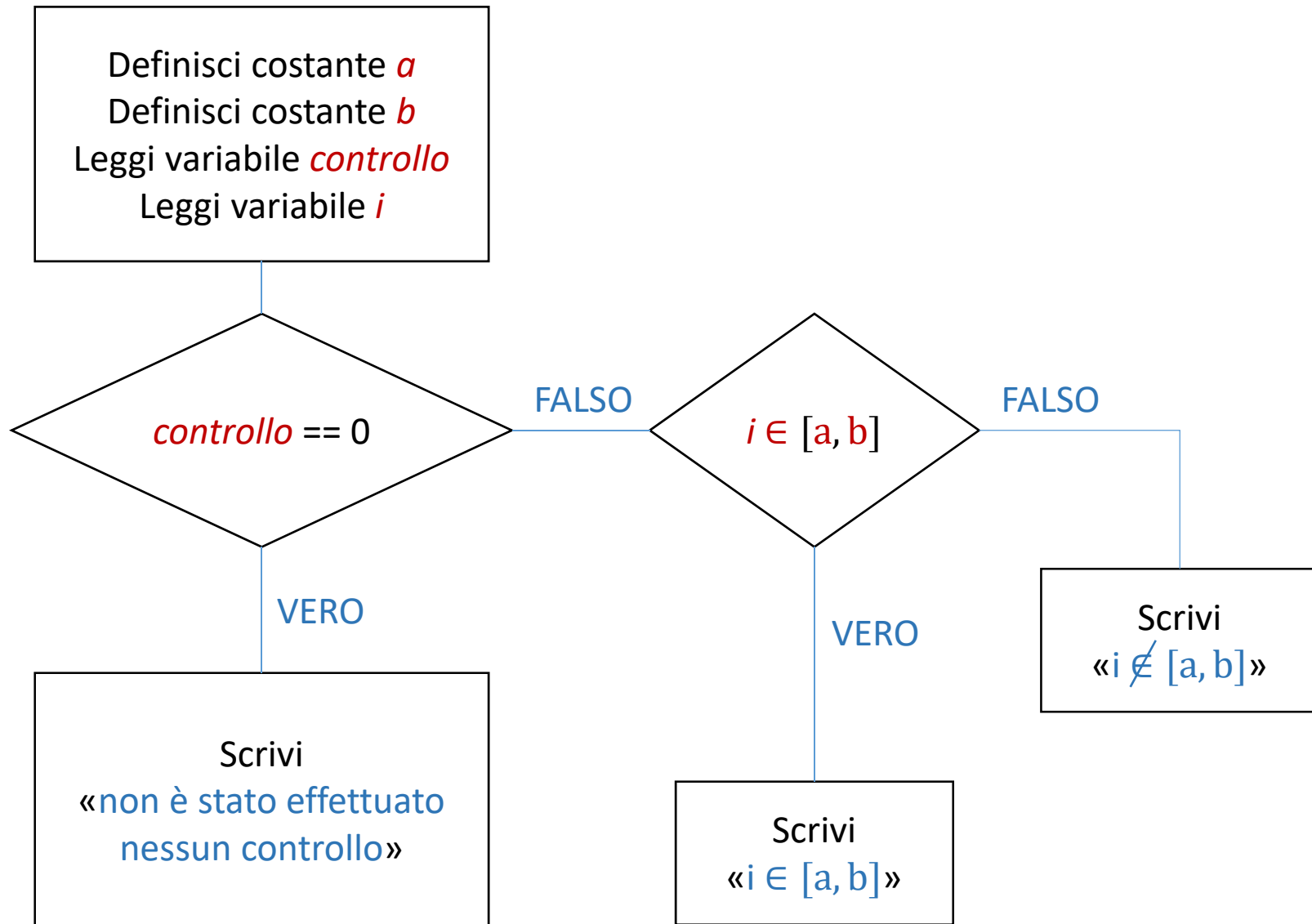
- legga da tastiera un intero positivo *numero* compreso tra 1 e 5
- stampi su schermo il numero ordinale corrispondente (primo, secondo, terzo, quarto, quinto)
- stampi un messaggio d'errore se il numero non è compreso tra 1 e 5
  
- VINCOLO
- Scrivere due varianti del programma: una che usa `if-else` annidati e una che usa `switch-case`

# Esercizio (2)

- Scrivere un programma che definisca ed inizializzi due **costanti** intere *a* e *b*, poi
- legga in ingresso due numeri interi: *controllo* ed *i*
- L'intero *controllo* si utilizza per controllare il comportamento del programma
  - se *controllo* è diverso da 0
    - il programma scrive un messaggio se *i* non è compreso nell'intervallo [*a*, *b*]
  - altrimenti segnala che non è stato effettuato nessun controllo

- **VINCOLO:** Non utilizzare l'operatore logico **&&**

# Diagramma di flusso



# Espressioni logiche

- Come verificare che  $i \in [a, b]$  senza utilizzare l'operatore logico `&&`?

`(i >= a) && (i <= b)`

- Ogni espressione di tipo `&&` si può verificare come una combinazione di NOT (!) e OR (||)
- Per verificare la **non** appartenenza all'intervallo

`!(i >= a) || !(i <= b) → (i < a) || (i > b)`

- E dunque l'appartenenza all'intervallo è verificata dall'espressione

`!((i < a) || (i > b))`

# Soluzione

```
main()
{
    const int a=10, b=100;
    int controllo, i;

    cout<<"\nInserisci un valore di controllo:";
    cin>>controllo;
    cout<<"\nInserisci un valore per i:";
    cin>>i;

    if (controllo==0)
        cout<<"\nNon è stato effettuato nessun controllo\n";
    else
        if ( !((i<a) || !(i>b)) )
            cout<<"\ni è dentro l'intervallo.\n";
        else
            cout<<"\ni è fuori dall'intervallo.\n";
}
```

**NOTA: Non servono le parentesi graffe perché il costrutto if/else è considerato una singola istruzione**



# Esercizio per casa

- Scrivere un programma **calcolatrice** che legga da tastiera due variabili intere **a** e **b**, poi
- Stampi un menu con un valore numerico associato a quattro operazioni possibili:
  1. *Addizione*
  2. *Sottrazione*
  3. *Moltiplicazione*
  4. *Divisione*
- legga in ingresso la variabile intera **controllo** che rappresenta la scelta di operazione
- Utilizzi un costrutto **switch** per effettuare l'operazione scelta usando **a** e **b** come operandi
- Stampi il risultato dell'operazione su schermo
  - Se l'operazione è una divisione bisognerà stampare parte intera e resto