



TANKERINO.com

Schemi di flusso

Benvenuti alla lezione di oggi! Oggi ci concentreremo su come affrontare, comprendere e risolvere un esercizio di programmazione. Prendiamo come esempio un esercizio che ci chiede di prendere in input due numeri e di eseguire alcune operazioni condizionali su di essi

- Chiedi 2 numeri all'utente.
- Se il primo è maggiore di 10 scrivi "FUORI SCALA".
- Se il secondo numero è maggiore di 50 scrivi "TROPPO GRANDE".
- Se non hai stampato nulla allora stampa "TUTTO OK"

1. Analisi dell'Esercizio

Il primo passo nella risoluzione di qualsiasi problema di programmazione è comprendere esattamente cosa ci viene chiesto. Dobbiamo leggere attentamente l'enunciato e identificare le operazioni chiave e le condizioni da verificare.

Nel nostro esercizio, ci viene chiesto di prendere in input due numeri e di verificare alcune condizioni su di essi, come se il primo numero è minore di 10 o se il secondo numero è maggiore di 50.

2. Pianificazione della Soluzione

Una volta compreso l'enunciato, dobbiamo iniziare a pensare a come affrontare il problema. Una buona pratica è quella di spezzare il problema in piccoli passi o operazioni che possiamo eseguire uno alla volta.

Prima di scrivere qualsiasi codice, è sempre una buona idea pianificare la soluzione e decidere l'ordine delle operazioni.

3. Creazione del Flowchart

Un ottimo strumento per visualizzare la logica del nostro codice è il diagramma di flusso o "flowchart". Questo ci aiuta a visualizzare l'ordine delle operazioni e le decisioni che il nostro programma dovrà prendere.

Nel nostro esercizio, il flowchart inizia con la richiesta dei due numeri, segue con le verifiche delle condizioni e termina con le azioni da eseguire in base ai risultati di queste verifiche.

4. Testare la Soluzione

Una volta che abbiamo una soluzione pianificata e un flowchart per visualizzarla, possiamo iniziare a scrivere il nostro codice. Ma non appena abbiamo finito di scrivere, dobbiamo testare il nostro codice per assicurarci che funzioni come previsto.

È essenziale testare il nostro codice con vari input per assicurarci che funzioni in tutti i casi possibili.

Nel nostro esercizio, potremmo testare il nostro codice inserendo numeri come 5 e 45, o 15 e 55, e vedere se otteniamo gli alert previsti.

Risoluzione

1. Input dei Numeri:

- Il programma inizia chiedendo all'utente di inserire il primo numero. Questo valore viene convertito in un numero intero e memorizzato nella variabile `primoNumero`.
- Successivamente, il programma chiede all'utente di inserire il secondo numero. Anche questo valore viene convertito in un numero intero e memorizzato nella variabile `secondoNumero`.

2. Verifica delle Condizioni:

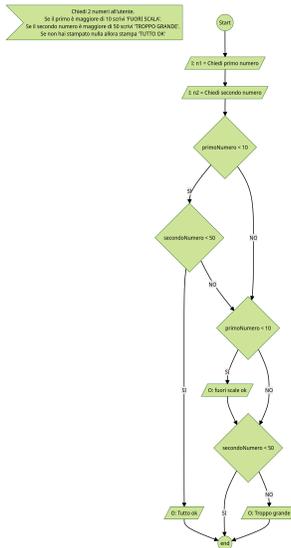
- Il programma verifica se `primoNumero` è minore di 10.
 - Se questa condizione è vera, il programma verifica ulteriormente se `secondoNumero` è minore di 50.
 - Se anche questa condizione è vera, viene mostrato un messaggio di alert con il testo "TUTTO OK".
- Successivamente, il programma verifica se `primoNumero` è maggiore di 10.
 - Se questa condizione è vera, viene mostrato un messaggio di alert con il testo "FUORI SCALA".
- Infine, il programma verifica se `secondoNumero` è maggiore di 50.
 - Se questa condizione è vera, viene mostrato un messaggio di alert con il testo

"TROPPO GRANDE".

3. Conclusione:

- Dopo aver eseguito tutte le verifiche, il programma termina la sua esecuzione.

In sintesi, il programma prende in input due numeri e, in base ai valori inseriti, esegue una serie di controlli condizionali per determinare quali messaggi di alert mostrare all'utente.



Codice Mermaid

```
graph TD
    Start((Start)) --> N1[/n1 = Chiedi primo numero/]
    N1 --> N2[/n2 = Chiedi secondo numero/]
    N2 --> D1{primoNumero < 10}
    D1 -- SI --> F1[/O fuori scala/]
    D1 -- NO --> D2{secondoNumero < 50}
    D2 -- SI --> F2[/O Troppo grande/]
    D2 -- NO --> F3[/O Tutto ok/]
    D1 -- NO --> F3
    F1 --> End((End))
    F2 --> End
    F3 --> End
```

```
start((Start))
input_n1[/I: n1 = Chiedi primo numero/]
input_n2[/I: n2 = Chiedi secondo numero/]
check_n1{primoNumero < 10}
check_n2{secondoNumero < 50}
print_all_ok[/O: Tutto ok/]
end
```

```
check_n1_2{primoNumero < 10}
print_fuori_scala[/O: fuori scale ok/]
end
```

```
check_n2_2{secondoNumero < 50}
end
```

```
print_troppo_grande[/0: Troppo grande/]
```

```
program_end((end))
```

```
start -->  
input_n1 -->  
input_n2 -->  
check_n1 -->|SI| check_n2  
check_n1 -->|NO| check_n1_2  
check_n2 -->|SI| print_all_ok  
print_all_ok --> program_end  
check_n2 -->|NO| check_n1_2
```

```
check_n1_2 -->|SI| print_fuori_scala  
check_n1_2 -->|NO| check_n2_2  
print_fuori_scala --> check_n2_2
```

```
check_n2_2 -->|SI| program_end  
check_n2_2 -->|NO| print_troppo_grande  
print_troppo_grande --> program_end
```

Conclusione

Risolvere esercizi di programmazione può sembrare complicato all'inizio, ma con pratica e metodo, diventerà sempre più semplice. Ricordatevi sempre di leggere attentamente l'enunciato, pianificare la vostra soluzione, visualizzarla con un flowchart e testare il vostro codice. Buona programmazione a tutti!

(CC BY-NC-SA 3.0) lezione - by tankerino.com

<https://www.tankerino.com>

Questa lezione e' stata realizzata grazie al contributo di:



Risorse per la scuola



Siti web a Varese

<https://www.baobab.school>

<https://www.francescobelloni.it>