



ESERCITAZIONE

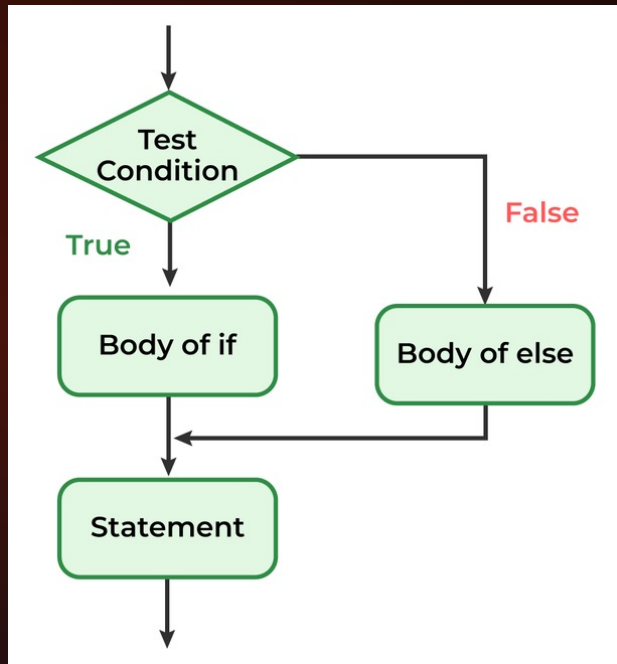
PARTE 2



`IF(){} ELSE{}`

CONTROLLIAMO DELLE CONDIZIONI

SE / ALTRIMENTI



In C, le istruzioni if e else servono a controllare il flusso di esecuzione del programma. Con if si verifica una condizione: se è vera, viene eseguito un blocco di codice. Se la condizione è falsa, l'istruzione else (opzionale) consente di eseguire un altro blocco di codice alternativo.

```
if (condizione) {
```

```
// Codice da eseguire se la condizione è vera
```

```
} else {
```

```
// Codice da eseguire se la condizione è falsa }
```

STRUTTURA

- `if (a > b) {`
- `printf("a è maggiore di b");`
- `} else {`
- `printf("a non è maggiore di b");`
- `}`



FOR E WHILE

FOR

Il ciclo for viene utilizzato per iterare un blocco di codice un numero definito di volte.

```
for (inizializzazione; condizione;  
    incremento/decremento) {  
    // Codice da eseguire ad ogni  
    iterazione  
}
```

ESEMPIO

```
for (int i = 0; i < 5; i++) {  
    printf("Valore di i: %d\n", i);  
}
```

WHILE

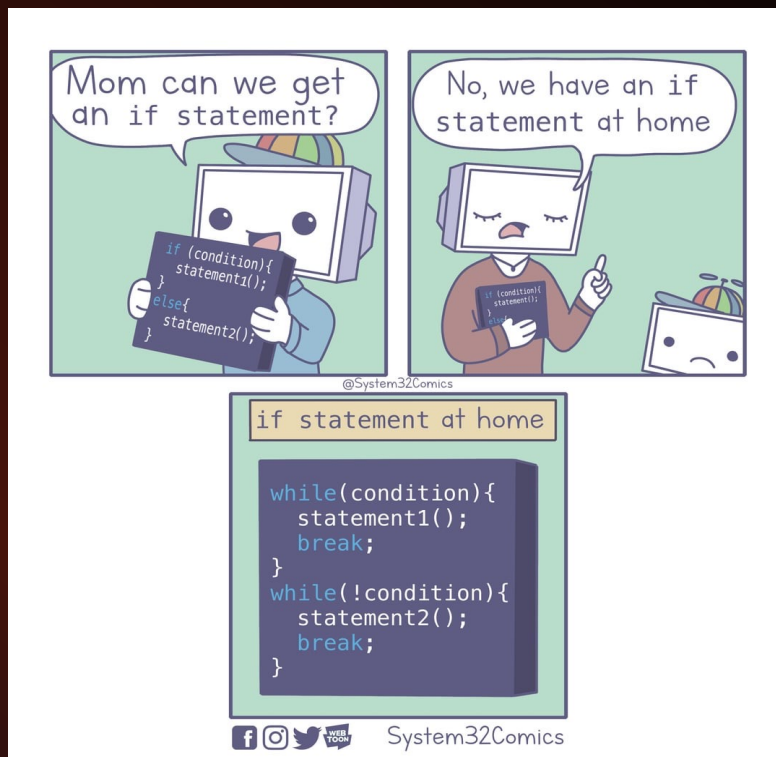
Il ciclo while esegue un blocco di codice finché una condizione rimane vera.

```
while (condizione) {  
    // Codice da eseguire finché la  
    condizione è vera  
}
```

ESEMPIO

```
int i = 0;  
while (i < 5) {  
    printf("Valore di i: %d\n", i);  
    i++; // Incremento di i per evitare ciclo  
        infinito  
}
```


MEME TIME



PROGRAMMERS: Know the difference!

```
() => {
  function getName(gender: String = ""): String {
    if(gender == "male") {
      return "Robert";
    }
    if(gender == "female") {
      return "Margot";
    }
    return "";
  }

  const gender = "male";
  const name = getName(gender);

  console.log('The child's name is ${name}');
})();
```

If statement



Switch case

ProgrammerHumor.io

CALCOLATRICE

«PROTOTIPO 2»

PROBLEMA

CREA UNA PICCOLA CALCOLATRICE
PRENDENDO IN INPUT 3 VARIABILI. SE $A = 0$
FAI LA SOMMA, ALTRIMENTI SE UGUALE A 1
LA SOTTRAZIONE

CODICE

```
#include <stdio.h>

int main() {

    int A, b, c, risultato;

    // Input da tastiera

    printf("Inserisci 0 per somma o 1 per differenza:
    ");

    scanf("%d", &A);

    printf("Inserisci il valore di b: ");

    scanf("%d", &b);

    printf("Inserisci il valore di c: ");

    scanf("%d", &c);
```

```
    // Controllo dell'operazione

    if (A == 0) {

        risultato = b + c;

        printf("La somma di %d e %d è: %d\n", b, c,
        risultato);

    } else if (A == 1) {

        risultato = b - c;

        printf("La differenza tra %d e %d è: %d\n", b, c,
        risultato);

    } else {

        printf("Operazione non valida. Inserisci 0 per
        somma o 1 per differenza.\n");    }

    return 0;}
```



BREAK

15 MINUTI

CALCOLIAMO IL FATTORIALE

SENZA USARE LA LIBRERIA 😊

WHILE

```
#include <stdio.h>
```

```
int main() {
```

```
int numero, fattoriale = 1;
```

```
// Input da tastiera
```

```
printf("Inserisci un numero intero positivo: ");  
scanf("%d", &numero);
```

```
// Controllo se il numero è negativo
```

```
if (numero < 0) {
```

```
printf("Il fattoriale non è definito per numeri  
negativi.\n"); }
```

```
else {
```

```
int i = 1;
```

```
// Inizializzazione del contatore while (i <=  
numero) {
```

```
fattoriale *= i;
```

```
// Calcolo del fattoriale
```

```
i++; // Incremento del contatore }
```

```
printf("Il fattoriale di %d è: %d\n", numero,  
fattoriale); }
```

```
return 0;}
```


FOR

```
#include <stdio.h>
```

```
int main() {
```

```
int numero, fattoriale = 1;
```

```
// Input da tastiera
```

```
printf("Inserisci un numero intero positivo: ");
```

```
scanf("%d", &numero);
```

```
// Controllo se il numero è negativo
```

```
if (numero < 0) {
```

```
printf("Il fattoriale non è definito per numeri  
negativi.\n"); }
```

```
else {
```

```
// Ciclo for per calcolare il fattoriale
```

```
for (int i = 1; i <= numero; i++) {
```

```
fattoriale *= i; // Calcolo del fattoriale }
```

```
printf("Il fattoriale di %d è: %d\n", numero,  
fattoriale); }
```

```
return 0;}
```



METTIAMO IN ORDINE

PROBLEMA

- Prendi in input 5 voti dell'esame di Programmazione e mettili in ordine crescente

SOLUZIONE

```
#include <stdio.h>

#define NUM_VOTI 5

int main() {

    int voti[NUM_VOTI]; // Array per memorizzare i voti

    int i, j, minIndex, temp; // Input dei voti

    for (i = 0; i < NUM_VOTI; i++) {

        printf("Inserisci il voto %d: ", i + 1);

        scanf("%d", &voti[i]);    }
```

```
// Ordinamento dei voti (Selection Sort)

for (i = 0; i < NUM_VOTI - 1; i++) {

    minIndex = i; // Assume che il primo elemento non ordinato sia il minimo

    for (j = i + 1; j < NUM_VOTI; j++) {

        if (voti[j] < voti[minIndex]) {

            minIndex = j; // Trova il nuovo minimo        }    }

        // Scambia il minimo trovato con il primo elemento non ordinato

        if (minIndex != i) {

            temp = voti[i];

            voti[i] = voti[minIndex];

            voti[minIndex] = temp;        }    }

        // Stampa dei voti ordinati

        printf("\nVoti in ordine crescente:\n");

        for (i = 0; i < NUM_VOTI; i++) {        printf("%d\n", voti[i]);    }    return 0;}
```

GRAZIE RAGAZZI
PER
L'ATTENZIONE😊

