



Operatori condizionali

Loris Giulivi – Nicolò Folloni

Recap

Le librerie contengono funzioni standard che possiamo usare nel nostro programma

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

```
void main()
{
```

```
    int num1;
    int num2;
```

```
    printf("Insert the first integer: ");
    scanf("%d", &num1);
    printf("Insert the second integer: ");
    scanf("%d", &num2);
```

```
    int risultato = num1 + num2;
    printf("The sum is: %d", risultato);
```

```
}
```

La funzione **main** è il punto di partenza del nostro programma

Le variabili contengono i dati e hanno un tipo specifico

Con **scanf** possiamo ottenere dati dall'utente e salvarli nelle variabili.

Possiamo stampare a schermo utilizzando **printf**. Possiamo stampare anche il valore delle variabili.

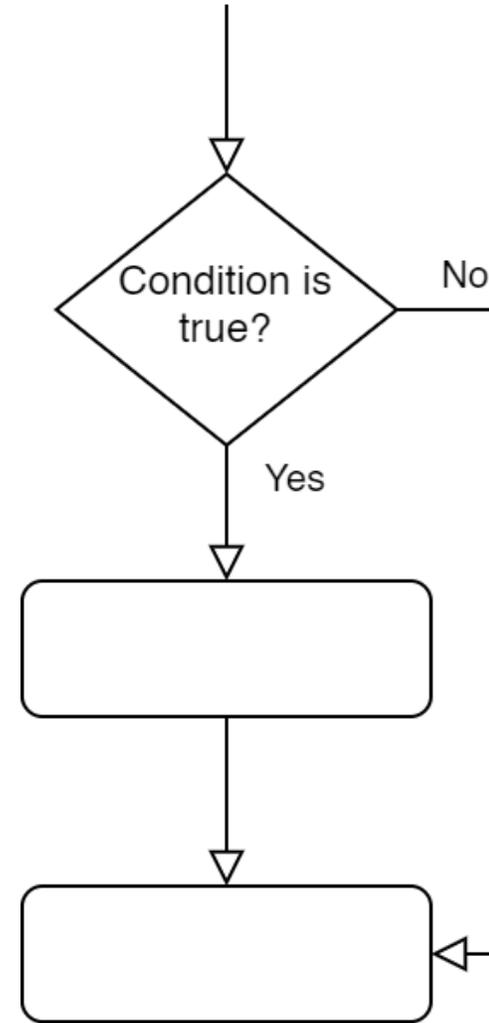
Operatore condizionale (1)

Devviare il flusso di esecuzione - **if**

Spesso vogliamo che certe porzioni di codice vengano eseguite solo sotto certe condizioni.

L'operatore condizionale **if** permette di eseguire un blocco di codice solo se una certa condizione è vera.

```
if (<condizione>)  
{  
    //DO SOMETHING  
}
```



Operatore condizionale (2)

Valori di verità

Il codice nel blocco compreso tra le graffe è eseguito solo se la condizione è *vera*.

In C, *vero* significa diverso da zero. Tipicamente, si indica la verità con il valore **1**, ma ciò non è obbligatorio.

```
int num1=7;
int num2=0;

if (num1)
{
    printf("This will be printed");
}

if (num2)
{
    printf("This will not");
}
```

Operatori logici e relazionali

Condizioni complesse

Per utilizzare condizioni più complesse, ci affidiamo agli operatori **logici** e **relazionali**, che ritornano un valore di verità {0, 1}.

<code>==</code>	uguale a
<code>!=</code>	diverso da
<code>></code>	maggiore di
<code><</code>	minore di
<code>>=</code>	maggiore o uguale a
<code><=</code>	minore o uguale a

<code>&&</code>	AND logico
<code> </code>	OR logico
<code>!</code>	NOT logico

```
int num1=7;
int num2=0;

if (num1>num2)
{
    printf("This will be printed");
}

if (!num2)
{
    printf("This will also be printed");
}
```

Esercizio 1

ES1: Scrivere un programma che chiede all'utente due numeri interi, e che stampa il più grande tra i due.

Esercizio 1 - Soluzione

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

void main()
{
    int num1;
    int num2;

    printf("First number: ");
    scanf("%d", &num1);

    printf("Second number: ");
    scanf("%d", &num2);

    if (num1 > num2)
    {
        printf("%d", num1);
    }

    if (num1 <= num2)
    {
        printf("%d", num2);
    }
}
```

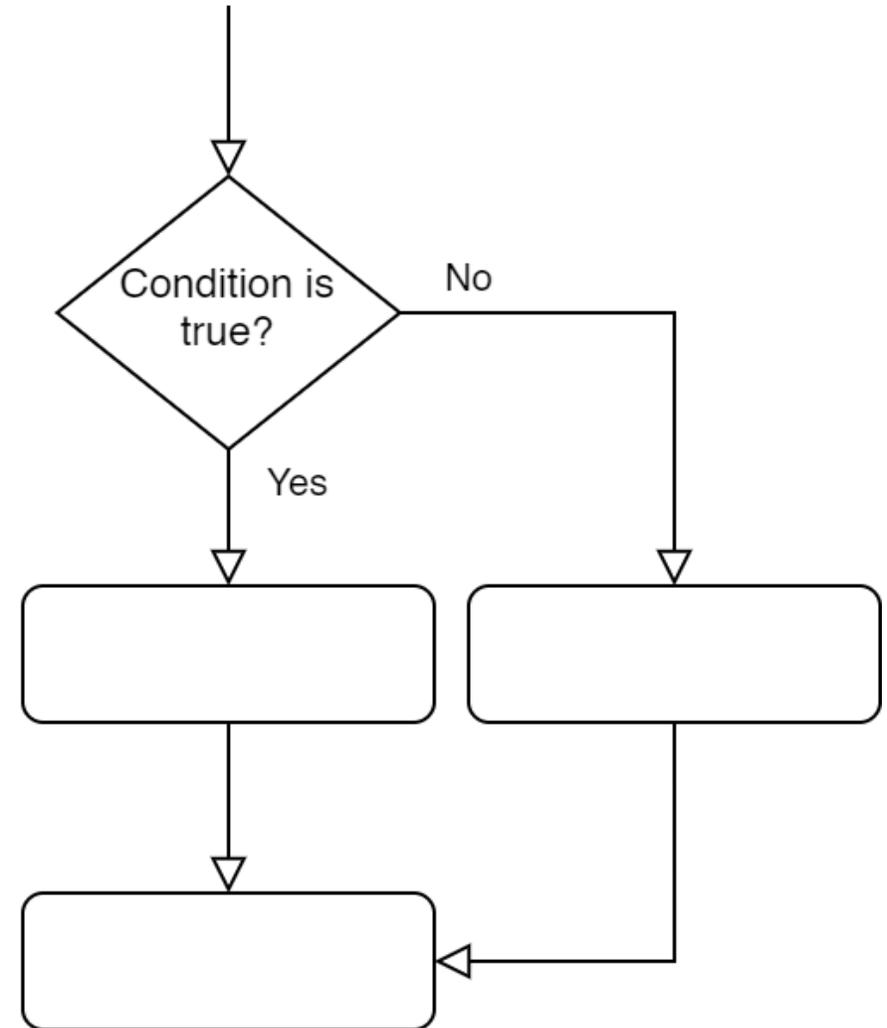
Possiamo fare meglio??

Operatore condizionale (3)

if, else

L'operatore **if** può essere esteso per comprendere vari rami del flusso di esecuzione, condizionatamente a più di una condizione.

```
if (<condizione>
{
    //DO SOMETHING
}
else
{
    //DO SOMETHING
}
```



Esercizio 1 – Soluzione con if-else

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

void main()
{
    int num1;
    int num2;

    printf("First number: ");
    scanf("%d", &num1);

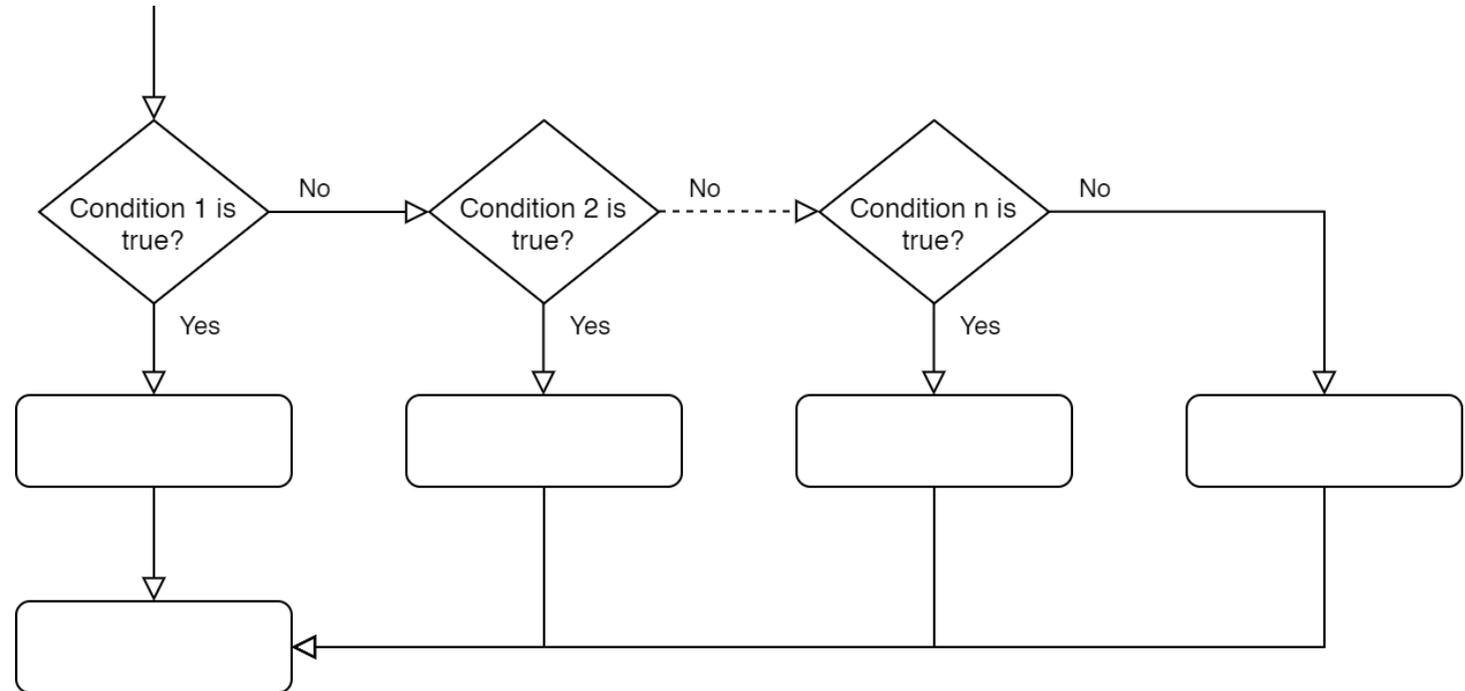
    printf("Second number: ");
    scanf("%d", &num2);

    if(num1>num2)
    {
        printf("%d", num1);
    }
    else
    {
        printf("%d", num2);
    }
}
```

Operatore condizionale (3)

if, else if, else

```
if(<condizione 1>
{
    //DO SOMETHING
}
else if(<condizione 2>)
{
    //DO SOMETHING
}
...
else if(<condizione n>)
{
    //DO SOMETHING
}
else
{
    //DO SOMETHING
}
```



Operatore condizionale (3)

⚠ Solo un ramo in una catena di **if**, **else if**, **else** può essere eseguito. Anche se più condizioni sono verificate, solo il ramo corrispondente alla **prima** condizione vera sarà eseguito.

```
int num1 = -2;
int num2 = 4;

if (num1 > 0)
{
    //This is not executed
}
else if (num2 > num1)
{
    //This is executed
}
else
{
    //This is not executed
}
```

Esercizio 2

ES2: Scrivere un programma che chiede in input due numeri interi positivi e stabilisce se sono oppure no uno il multiplo dell'altro.

Esercizio 2 - Soluzione

```
int a;
int b;

printf("Enter the first number: ");
scanf("%d", &a);
printf("Enter the second number: ");
scanf("%d", &b);

if(a<=0 || b<=0)
{
    printf("Wrong input. Please enter positive numbers only");
}
else if(a%b == 0)
{
    printf("%d is a multiple of %d", a, b);
}
else if(b%a == 0)
{
    printf("%d is a multiple of %d", b, a);
}
else
{
    printf("%d and %d are not multiples", a, b);
}
```

Operatore condizionale annidato

How deep can you go?

Gli operatori condizionali (e tutti quelli che vedremo nelle prossime lezioni) possono essere inseriti anche annidati dentro altri blocchi di codice.

```
int num1 = 2;
int num2 = 4;

if(num1+1 > num2)
{
    //This is not executed because
    //3 > 4 False
}
else
{
    if(num1 < num2)
    {
        if(num1+3 > num2)
        {
            //This is executed because
            //5 > 4 True
            //(And because 2<4)
        }
    }
    else
    {
        //This is not executed because
        //the 'if' branch was executed
    }
}
```

Esercizio 3

ES3: Quali rami del codice vengono eseguiti?

```
int num1 = 4;
int num2 = 2;
int num3 = 9;

if (num1 > num2)
{
    if (! (num1 > num2))
    {
        //A
    }
    else
    {
        //B
    }

    if (num1 && num2)
    {
        //C
    }
}
else if (num3 > num2)
{
    //D
}
else
{
    if (num1)
    {
        //E
    }
}
```

Esercizio 3 - Soluzione

ES3: Quali rami del codice vengono eseguiti?

- A** No, **!(num1 > num2)** è falso
- B** Sì, perché **A** non è stato eseguito
- C** Sì, perché **num1 != 0** e **num2 != 0**
- D** No, perché l'altro ramo è stato eseguito
- E** No, perché l'altro ramo è stato eseguito

```
int num1 = 4;
int num2 = 2;
int num3 = 9;

if (num1 > num2)
{
    if (!(num1 > num2))
    {
        //A
    }
    else
    {
        //B
    }

    if (num1 && num2)
    {
        //C
    }
}
else if (num3 > num2)
{
    //D
}
else
{
    if (num1)
    {
        //E
    }
}
```

Esercizio 4

ES4: Scrivere un programma che presi in input due numeri interi positivi o nulli indichi all'utente se il loro rapporto è maggiore, minore o uguale a 1.

CHALLENGE: risolvere senza far calcolare al programma il rapporto tra i numeri.

Esercizio 4 - Soluzione

1

```
int a;  
int b;  
  
printf("Enter first number: ");  
scanf("%d", &a);  
printf("Enter second number: ");  
scanf("%d", &b);
```

Per far sì che si ottenga il rapporto come numero decimale, è necessario che le variabili vengano convertite in float, altrimenti l'operazione viene eseguita tra interi, e le parti decimali saranno troncate.

2

```
if (a < 0 || b < 0)  
{  
    printf("Please, enter non-negative numbers only");  
}  
else if(b==0)  
{  
    printf("Cannot divide by zero!");  
}  
else  
{  
    float c = (float)a/(float)b;  
  
    if(c>1.)  
    {  
        printf("Ratio is greater than 1");  
    }  
    else if(c==1)  
    {  
        printf("Ratio is equal to 1");  
    }  
    else  
    {  
        printf("Ratio is smaller than 1");  
    }  
}
```

Esercizio 4 – Soluzione (challenge)

1

```
int a;  
int b;
```

```
printf("Enter first number: ");  
scanf("%d", &a);  
printf("Enter second number: ");  
scanf("%d", &b);
```

2

```
if (a < 0 || b < 0)  
{  
    printf("Please, enter non-negative numbers only");  
}  
else if (b != 0)  
{  
  
    if (a > b)  
    {  
        printf("Ratio is greater than 1");  
    }  
    else if (a == b)  
    {  
        printf("Ratio is equal to 1");  
    }  
    else  
    {  
        printf("Ratio is smaller than 1");  
    }  
}  
else  
{  
    printf("You cannot divide by zero!");  
}
```

Esercizio 5

ES5: Scrivere un programma che chiede all'utente tre numeri interi, e che stampa i tre numeri in ordine decrescente.

Esercizio 5 - Soluzione

 Più variabili *dello stesso tipo* possono essere dichiarate nella stessa istruzione.

1

```
int num1, num2, num3;
```

```
printf("First number: ");
scanf("%d", &num1);
printf("Second number: ");
scanf("%d", &num2);
printf("Third number: ");
scanf("%d", &num3);
```

2

```
if(num1>num2)
{
    if(num2>num3)
    {
        printf("%d-%d-%d", num1, num2, num3);
    }
    else
    {
        if(num1>num3)
        {
            printf("%d-%d-%d", num1, num3, num2);
        }
        else
        {
            printf("%d-%d-%d", num3, num1, num2);
        }
    }
}
else
{
    if(num2>num3)
    {
        if(num1>num3)
        {
            printf("%d-%d-%d", num2, num1, num3);
        }
        else
        {
            printf("%d-%d-%d", num2, num3, num1);
        }
    }
    else
    {
        printf("%d-%d-%d", num3, num2, num1);
    }
}
```

Esercizio 6

ES6: Scrivere un programma che richiede in input la lunghezza dei tre lati di un triangolo. Verificare se i numeri inseriti possono effettivamente rappresentare i lati di un triangolo (suggerimento: cosa dice la disuguaglianza triangolare?). In caso negativo, visualizzare un messaggio di errore. In caso affermativo, stabilire se il triangolo è equilatero, isoscele o scaleno, e calcolarne il perimetro.

Esercizio 6 - Soluzione

1

```
int a;  
int b;  
int c;
```

```
printf("Inserire la lunghezza del primo lato: ");  
scanf(" %d", &a);  
printf("Inserire la lunghezza del secondo lato: ");  
scanf(" %d", &b);  
printf("Inserire la lunghezza del terzo lato: ");  
scanf(" %d", &c);
```

2

```
if(a>b+c || b>a+c || c>a+b)  
{  
    printf("a, b e c non sono i lati di un triangolo\n");  
}  
else  
{  
    if(a == b && a == c)  
  
    {  
        printf("Il triangolo e' equilatero\n");  
    }  
    else if(a == b || b == c || a == c)  
    {  
        printf("Il triangolo e' isoscele\n");  
    }  
    else  
    {  
        printf("Il triangolo e' scaleno\n");  
    }  
  
    int p;  
    p = a+b+c;  
    printf("perimetro: %d\n", p);  
}
```