



UNIVERSITY OF TRENTO - Italy

Laboratorio 3

Pierluigi Roberti
Carmelo Ferrante

DISI – aa 2024-25

Università degli Studi di Trento

pierluigi.roberti@unitn.it

Da ricordare

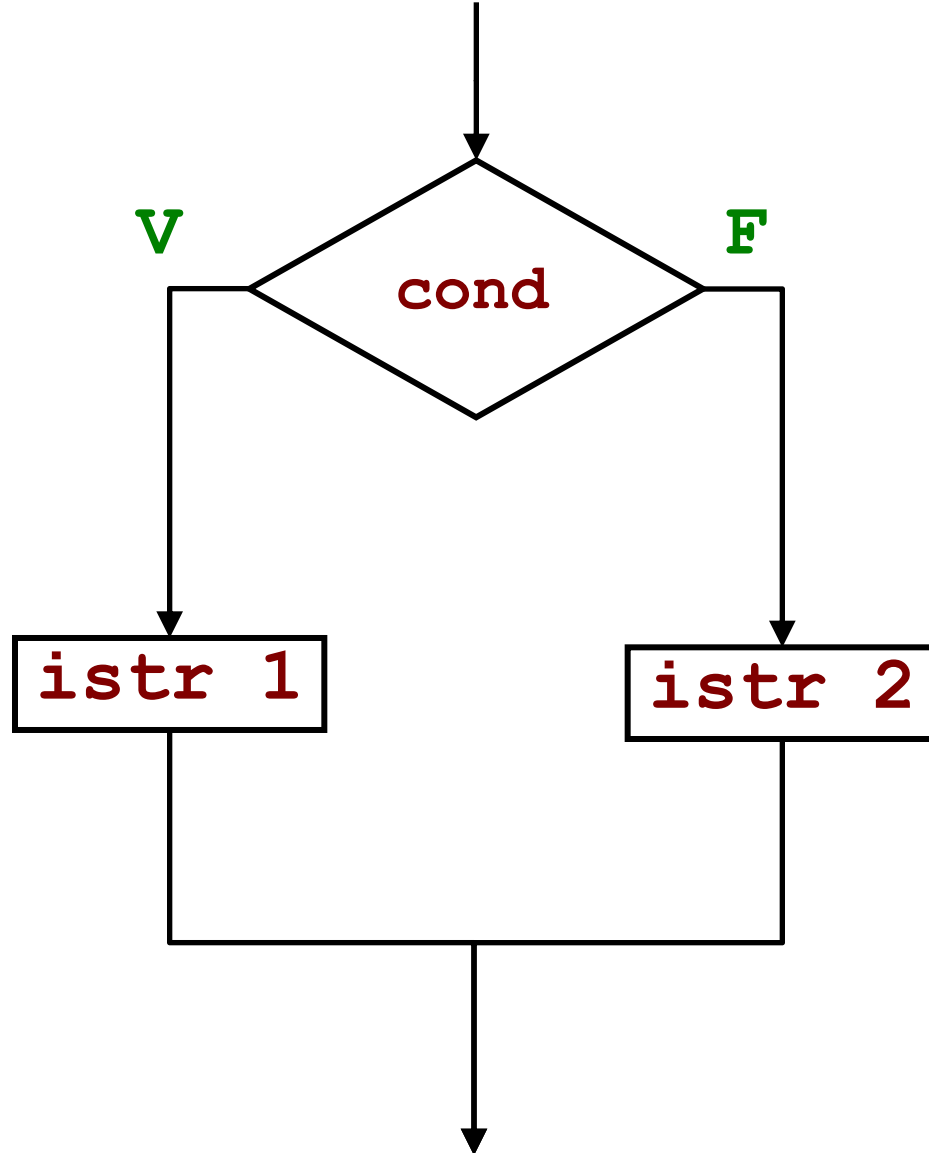
- **Costrutto if:**

```
<inizializzazione variabili>;  
if (<condizione>) {  
    <istruzioni1>;  
} else {  
    <istruzioni2>;  
}
```

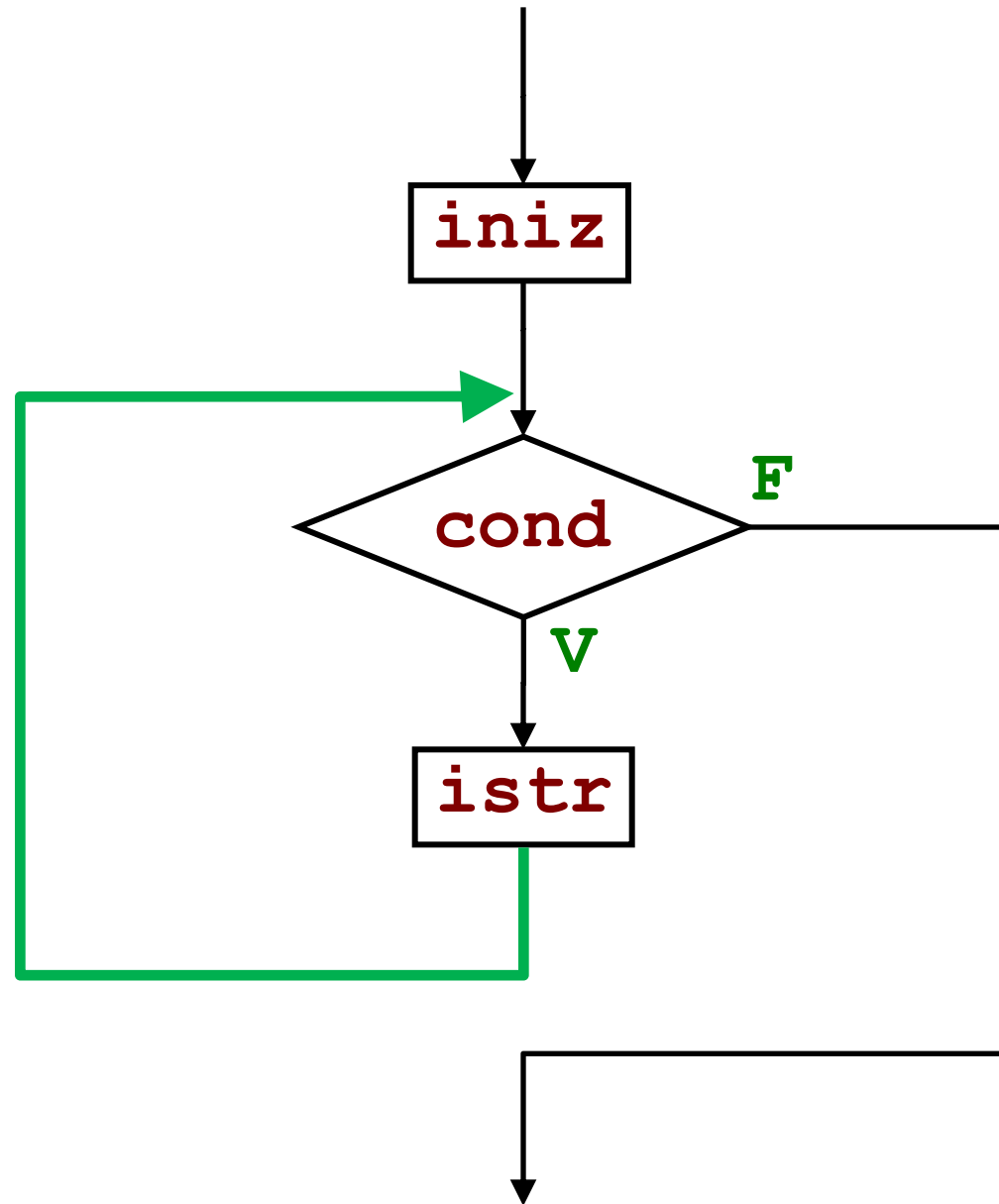
- **Costrutto while:**

```
<inizializzazione variabili>;  
while (<condizione>) {  
    <istruzioni>;  
}
```

Costrutto IF



Costrutto WHILE



Esercizio 1a

- Scrivere un programma che
 - Chiede all'utente un numero intero positivo
 - `printf(«...»); scanf(«%d», &x);`
 - Stampa a video la sequenza dei numeri naturali compresi tra 0 e un numero intero x inserito da tastiera (es. 0 1 2 3 4 ... x) `printf(«%d», i)`
 - quale è il valore della variabile di controllo quando il programma termina?
 - gli estremi vengono mostrati?

- **Da sapere/ricordare:**

```
<inizializzazione>;  
while (<condizione>) {  
    <istruzione>;  
    ...  
    <istruzione>;  
}
```

Esercizio 1b

- Scrivere un programma che
 - Chiede all'utente un numero intero positivo
 - Controllare che il valore di x sia compreso tra 0 e 10
 - Controllo input
 - Se valore non corrisponde alle richieste: richiedere il valore fino a che non soddisfa i requisiti
 - Stampa a video la sequenza dei numeri naturali compresi tra 0 e un numero intero x inserito da tastiera (es. 0 1 2 3 4 ... x)
 - quale è il valore della variabile di controllo quando il programma termina?
 - gli estremi vengono mostrati?

- **Da sapere/ricordare:**

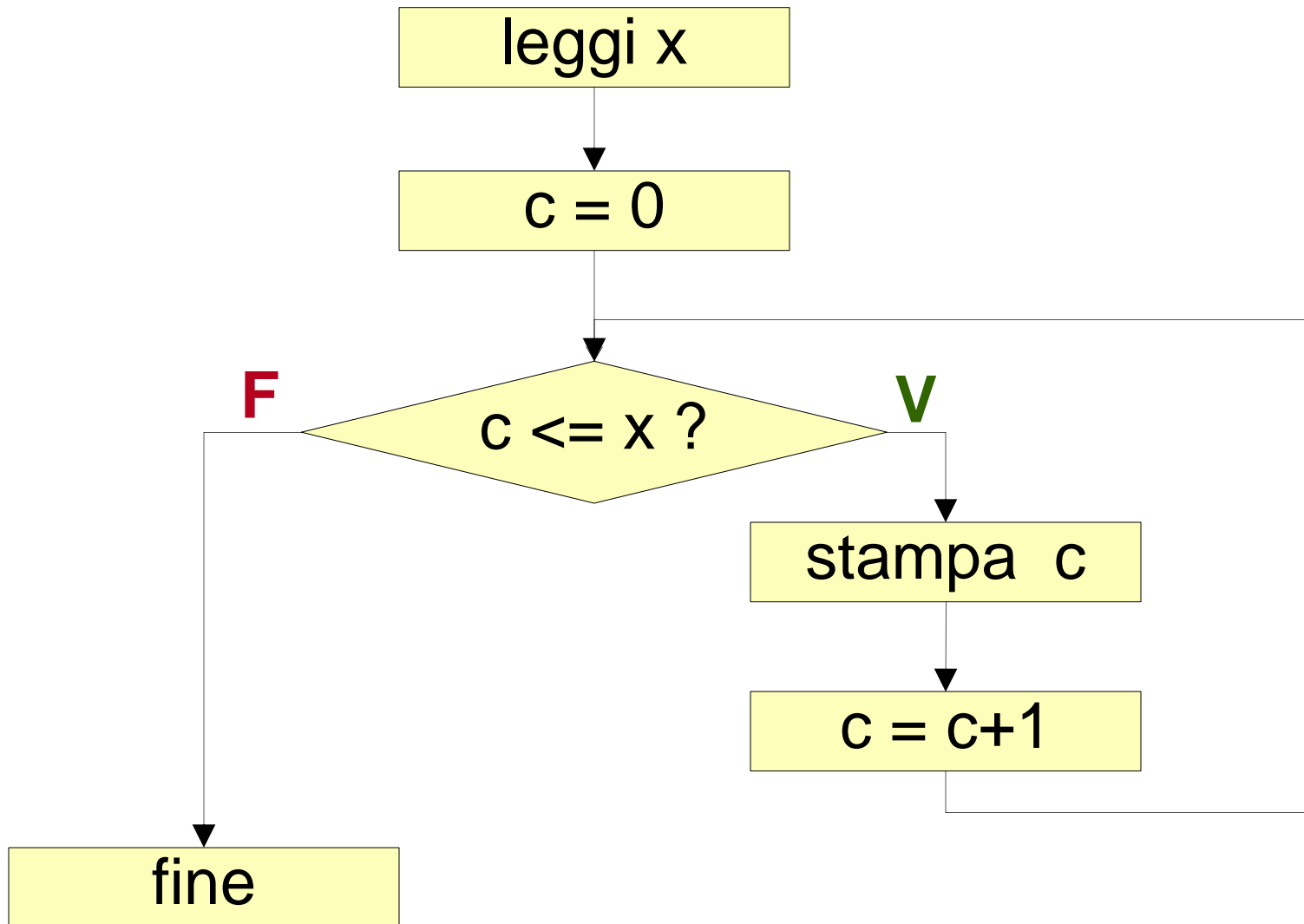
```
<istruzione>;  
while (<condizione>) {  
    <istruzione>;  
}
```

Esercizio 1 - Soluzione

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

```
int main() {
    int contatore, x;
    printf("x="); scanf("%d", &x);
    while (x<0 || x>10) {
        printf("Errore 0<=x<=10 \n");
        printf("x="); scanf("%d", &x);
    }
    contatore = 0;
    while (contatore <= x) {
        printf("%d ", contatore);
        contatore = contatore + 1;
    }
    printf("\n");
    printf("Ho finito di contare :-)\n");
    system("PAUSE");
}
```

Controllo
input utente



Combinare costrutti

- Una possibilità:

```
<inizializzazione var>;  
while (<condizione1>)  
{  
    <blocco1>;  
}
```

```
if (<condizione2>){  
    <istruzione1>;  
}else {  
    if (<condizione3>) {  
        <istruzione2>;  
    }else{  
        <istruzione3>;  
    }  
}
```

- Un'altra possibilità:

```
<inizializzazione var>;  
if (<condizione1>) {  
    <blocco1>;  
} else {  
    <blocco2>;  
}
```

```
while (<condizione1>){  
    <istruzione1>;  
}
```

- Si possono innestare **while** (o **if**) in **while** (o **if**) in **while** (o **if**) in ...

Esercizio 2

- Scrivere un programma che
 - Chiede all'utente un numero intero positivo x
 - Controllare che il valore di x sia compreso tra 0 e 10
 - Controllo input
 - Se valore non corrisponde alle richieste: richiedere il valore fino a che non soddisfa i requisiti
 - Se x MINORE di 5
 - Stampare a video i numeri compresi tra x e 10
 - Ordine crescente
 - Altrimenti (x MAGGIORE o UGUALE a 5)
 - Stampare a video i numeri compresi tra x e 0
 - Ordine decrescente

*Esempio di cicli annidati
in costruito if-else*

Esercizio 3a

- Scrivere un programma che
 - Chiede all'utente un numero intero positivo MAGGIORE di 1950
 - Considerare il numero inserito come un anno
 - Stampare a video tutti gli anni compresi tra 1900 e l'anno indicato
 - Ordine crescente
 - Estremi inclusi

- Opzionale:
 - Controllo input
 - Se valore non corrisponde alle richieste: richiedere il valore fino a che non soddisfa i requisiti

Esercizio 3b

- Scrivere un programma che
 - Chiede all'utente un numero intero positivo x MAGGIORE di 1950
 - Considerare x come un anno
 - Stampare a video tutti gli anni **BISESTILI** compresi tra 1900 e l'anno indicato
 - Ordine crescente
 - Estremi inclusi
- Opzionale:
 - Controllo input
 - Se valore non corrisponde alle richieste: richiedere il valore fino a che non soddisfa i requisiti

Esempio di costruito
if/if-else annidato in ciclo

Esercizio 3b

- Un anno è bisestile se
 - se è divisibile per 400 **oppure**
 - se è divisibile per 4 ma non per 100.

```
if (anno%400==0 || (anno%4==0 && anno%100!=0))  
{  
    printf("%d anno bisestile\n", anno);  
}
```

Nota – Esercizio 3b

- Operatore MODULO: %
 - Resto della divisione
 - Solo tra valori interi (int)

$$3 \% 2 \rightarrow 1$$

$$10 \% 3 \rightarrow 1$$

$$6 \% 4 \rightarrow 2$$

$$9 \% 5 \rightarrow 4$$

$$6 \% 3 \rightarrow 0$$

3 è un divisore di 6

Esercizio 4

- Scrivere un programma che legga 5 valori interi da tastiera e calcoli la media, il massimo e il minimo
 - quante variabili sono necessarie? di che tipo? come vengono inizializzate?
- **Da sapere/ricordare:**
 - la media si calcola sommando tutti i valori e dividendo la somma per il numero di valori
- **In aggiunta:**
 - inserire da tastiera il numero di valori da mediare

Nota – Esercizio 4b

- Utilizzare la logica vista nella ricerca del massimo tra il contenuto di tre variabili n1, n2, n3
 - Esercitazione 2
 - Utilizzare una variabile per contenere il valore massimo

```
max = n1;
```

Decisione arbitraria: ipotizzo che il primo valore sia il massimo

```
if(n2 > max){ max = n2; }
```

```
if(n3 > max){ max = n3; }
```

Valuto gli altri valori per vedere se la mia ipotesi è corretta

```
// variabile max contiene il valore massimo
```


Esercizio 5

- Modificare il programma dell'esercizio 1 perché stampi a video tutti i divisori di **x**
- Esempio:

Inserisci un numero intero: 20

Output: 1 2 4 5 10 20

- Controllare che l'input fornito dall'utente sia strettamente positivo
 - Nel caso in cui il valore inserito dall'utente sia errato, l'input deve essere rifiutato con un messaggio di errore, ed il dialogo di input riproposto

Esercizio 5 – algoritmo

Chiedere x all'utente

Fintanto che $x \leq 0$

Stampa «errore»

Richiedere x all'utente

Controllo
input utente

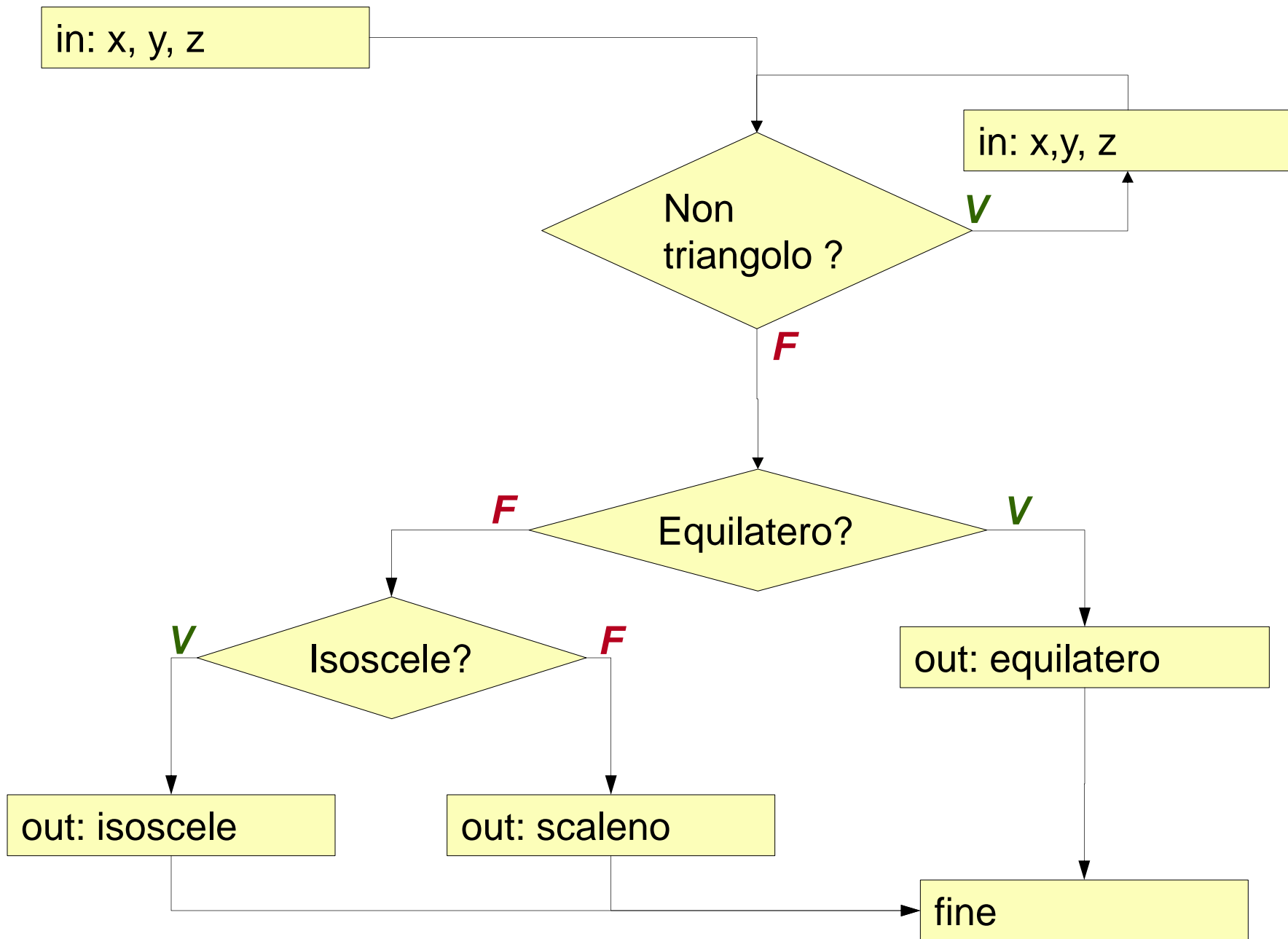
Per ogni valore i compreso tra 1 e x

Se i è un divisore di x

Stampare a video il numero i

Esercizio 6

- Scrivere un programma che richieda di inserire i valori dei 3 lati di un triangolo
- Controllare che siano dati validi:
 - siano strettamente positivi
 - che siano valori validi per un triangolo
- In caso di condizione precedente verificata verificare di che tipo di triangolo si tratta:
 - Equilatero, isoscele, scaleno
 - Soluzione vista in precedente esercitazione



Esercizio 6 – Soluzione

```
int x, y, z;
printf("lato x: "); scanf("%d", &x);
printf("lato y: "); scanf("%d", &y);
printf("lato z: "); scanf("%d", &z);

//while( x<=0 || y<=0 || z<=0 || x+y<z || x+z<y || y+z<x )
while(!( x>0 && y>0 && z>0 && x+y>z && x+z>y && y+z>x ))
{ printf("lato x: "); scanf("%d", &x);
  printf("lato y: "); scanf("%d", &y);
  printf("lato z: "); scanf("%d", &z);
}
/* i dati sono lati di un triangolo */

...
```

Esercizio 7a

- Chiedere all'utente due valori interi
 - li (limite inferiore)
 - ls (limite superiore)
- Calcolare la somma di tutti gli elementi compresi tra li e ls
 - Limiti li e ls inclusi
- Nota:
 - Anziché stampare a video i valori (tipo esercizio 1) si devono sommare i valori
 - Utilizzare una variabile per contenere la somma

Esercizio 7b

- Definizione alternativa dell'esercizio
- Scrivere un programma in grado di calcolare

$$S = \sum_{i=li}^{ls} i$$

dove li e ls sono inseriti dall'utente

- Nota:
 - Anziché stampare a video i valori (tipo esercizio 1) si devono sommare i valori
 - Utilizzare una variabile per contenere la somma

Esercizio 8

- Calcolare le seguenti sommatorie dove *li* e *ls* (ed ove richiesto *a*) sono inseriti dall'utente

$$\sum_{i=li}^{ls} i^3$$

$$\sum_{i=li}^{ls} (1/i)$$

$$\sum_{i=li}^{ls} (a+i)^3$$

$$\sum_{i=li}^{ls} (-1)^i * i$$

Nota – Esercizio 8

$$\sum_{i=li}^{ls} (-1)^i * i$$

- Se i pari
 - $(-1)^i \rightarrow +1 \rightarrow$ sommare valore i
- Se i dispari
 - $(-1)^i \rightarrow -1 \rightarrow$ sottrarre valore i
- Numero x pari
 - x divisibile per 2
 - resto dell'operazione $x / 2$ è zero
 - $x \% 2 \rightarrow 0$