



UNIVERSITY OF TRENTO - Italy

Laboratorio 4

Pierluigi Roberti
Carmelo Ferrante

DISI – aa 2024-25
Università degli Studi di Trento
pierluigi.roberti@unitn.it

Da ricordare

Costrutto if e while:

<init var>

```
if (<condizione>) {  
    <istruzione>;  
} else {  
    <istruzione>;  
}
```

<init var>;

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

Stampare a video

printf("[Testo][descrittori]", [variabili]);

Lettura input di utente da tastiera

scanf("[descrittore]", [indirizzo variabile]);

indirizzo variabile: **&***nome_variabile*

Esercizio Ripasso

- Chiedere all'utente due valori interi
 - li (limite inferiore)
 - ls (limite superiore)
- Calcolare la somma di tutti gli elementi compresi tra li e ls (inclusi)
- Vincoli: $li \geq 0 \ \&\& \ ls > li$

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

- Altra sommatoria: $\sum_{i=li}^{ls} i^3$
- Uso della produttoria: $\prod_{i=li}^{ls} i^3$

```
#include <math.h>
pow(base, esponente);
```

Operatore Ternario

L'operatore ternario "? :" è una forma sintetica dell'istruzione if-else, e per questo viene usata per ragioni di comodità e sinteticità del codice.

L'operatore "? :" è l'unico operatore ternario del C/C++, infatti opera su tre parametri a differenza degli altri operatori che spesso operano su uno o due parametri.

Qui di seguito presentiamo la sintassi dell'operatore e la corrispondente sintassi di un blocco if-else:

// Operatore ternario ?

x = condizione ? espressione1 : espressione2;

// Corrispondente blocco if-else

if (condizione) {

x = espressione1;

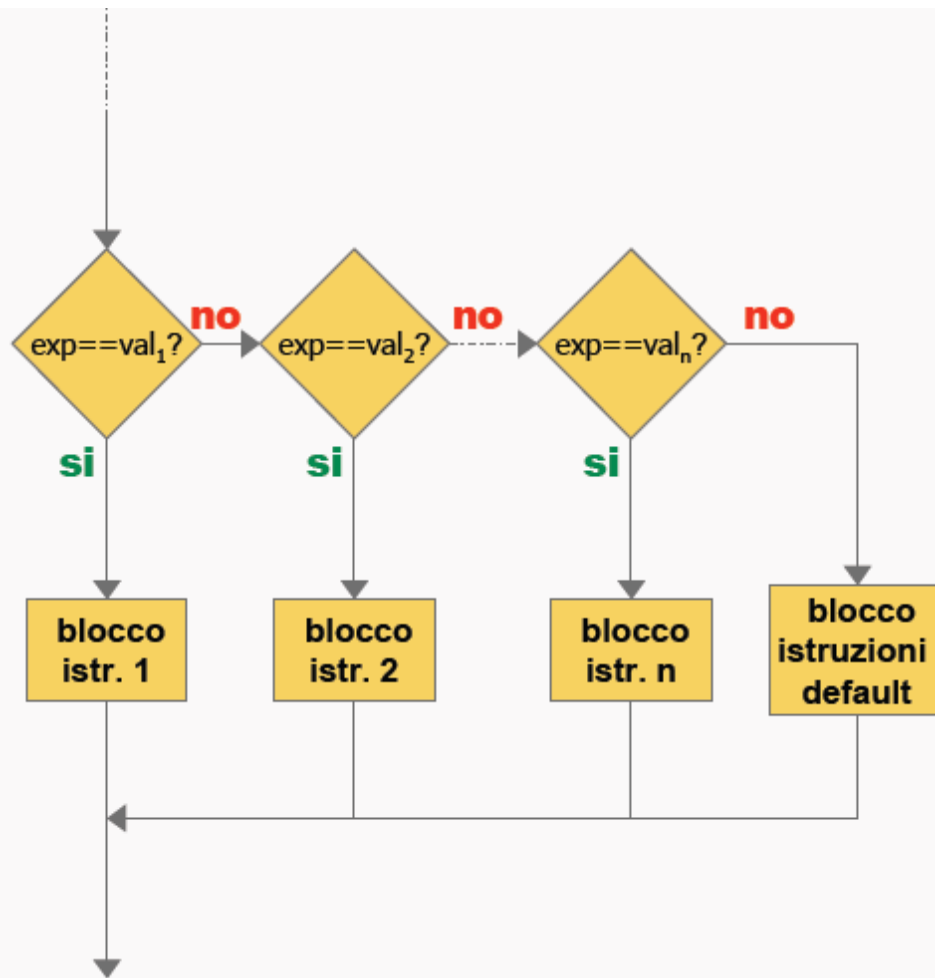
} else {

x = espressione2;

}

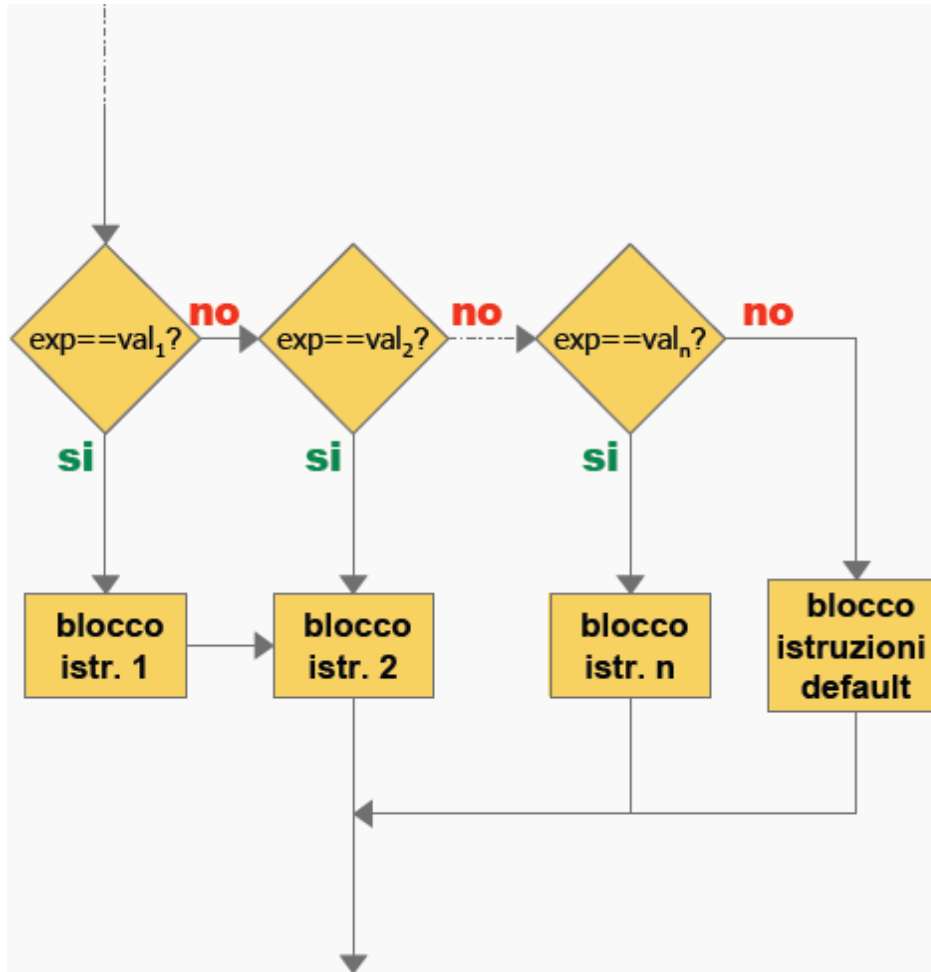
```
int x;  
printf("x="); scanf("%d",&x);  
//calcolo del valore assoluto: x= | x |  
x = ( x > 0 ) ? x : -x;
```

Il costrutto “switch”



```
...  
switch (espressione) {  
  
    case  $val_1$ : blocco istr. 1  
                break;  
  
    case  $val_2$ : blocco istr. 2  
                break;  
  
    ...  
  
    case  $val_n$ : blocco istr. n  
                break;  
  
    default:  
                blocco istr. default  
  
}  
...
```

Il costrutto “switch” senza “break”



```
...  
switch (espressione) {  
    case val1: blocco istr. 1  
    tolto break! →  
    case val2: blocco istr. 2  
                break;  
    ...  
    case valn: blocco istr. n  
                break;  
    default:  
                blocco istr. default  
}  
...
```

switch Statement

- L'istruzione **switch** ha delle differenze dall'if-else, infatti può essere usata solo in alcuni casi:
 - Viene valutata solamente una variabile, tutte le scelte dipenderanno, infatti, da questa variabile. La variabile deve essere un tipo **int**, **short**, **long** o **char**.
 - Ogni singolo valore della variabile può controllare solo una scelta. Una scelta finale, chiamata **default** è incaricata di catturare tutti i valori dei casi non specificati.

C++ switch Statement

- Struttura di selezione multipla (Sostituisce catene di if ...else)
- L'istruzione switch ha la seguente struttura (Sintassi):

```
switch( int_expression ) {  
    case valore1: {blocco istruzioni} break;  
    case valore2: {blocco istruzioni} break;  
    ....  
    default: {blocco istruzioni} break;  
}
```

Non
obbligatorio

- Semantica
 - Esegue il case con il *valore_x* pari a *int_expression* oppure il default
- Notiamo che lo switch ha, inoltre, una struttura abbastanza particolare, è a cascata, cioè se eseguo il primo caso (avendo sotto degli altri) e non metto un **break** per uscire, continua ad eseguire anche le istruzioni successive che fanno parte di altre casistiche.

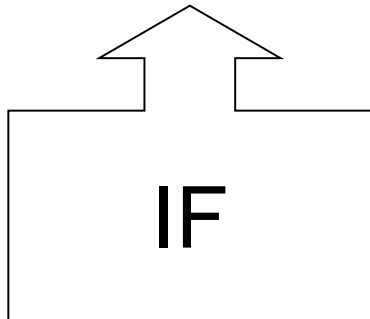
switch Statement - esempio

Ma vediamo l'esempio pratico, assumendo di avere la variabile intera "numero" che può assumere valori maggiori od uguali a zero:

```
switch (numero) {  
    case 0:  
        printf("Nessuno");  
        break;  
    case 1:  
        printf("Uno");  
        break;  
    case 2:  
        printf("Due");  
        break;  
    case 3:  
    case 4:  
    case 5: } 3, 4 o 5 sono in OR  
        printf("Valore positivo piccolo");  
        break;  
    default:  
        printf("Valore positivo grande");  
        break;  
}
```

switch Statement and if Statement

```
stato = stato +1;  
if (stato == 5) {stato=1; }  
if (stato == 1) {verso=1; }  
if (stato == 2) {verso=0; }  
if (stato == 3) {verso=-1;}  
if (stato == 4) {verso=0;}
```



A diagram representing a SWITCH statement. It consists of a rectangular box with the word "SWITCH" centered inside. Below the bottom edge of the box, there is a horizontal line with a small vertical step down on each side, and a diagonal line connecting the two steps, forming a downward-pointing arrow shape.

SWITCH

```
stato++; // stato = stato+1;  
if (stato==5) {stato=1;}  
switch (stato) {  
    case 1 : verso=1; break;  
    case 2 : verso=0; break;  
    case 3 : verso=-1; break;  
    case 4 : verso=0; break;  
}
```

Esercizio 1

Scrivere un programma che:

- Chiede all'utente un numero 0,1 oppure 2
- Stampa a video la scelta fatta oppure un messaggio di errore se il valore inserito non corrisponde a nessuno di quelli attesi

Creare 2 soluzioni nello stesso programma:

- una con l'utilizzo del costrutto if/if-else
- una con l'utilizzo del costrutto switch

Esercizio 1 – Soluzione parte a

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int scelta; printf("scelta="); scanf("%d",&scelta);
    /* opzione 1 prima versione */
    if(scelta==0){
        printf("0\n");
    } else {
        if (scelta==1){
            printf("1\n");
        } else {
            if (scelta==2){
                printf("2\n");
            } else {
                printf("errata\n") ;
            }
        }
    }
}
/* opzione 1 seconda versione - non ottimizzata! */
if (scelta==0){ printf("0\n"); }
if (scelta==1){ printf("1\n"); }
if (scelta==2){ printf("2\n"); }
if (scelta!=0 && scelta!=1 && scelta!=2){ printf("errata\n"); }
```

Esercizio 1 – Soluzione parte b

```
/* opzione 2 */  
switch (scelta){  
    case 0: printf("0\n"); break;  
    case 1: printf("1\n"); break;  
    case 2: printf("2\n"); break;  
    default: printf("errata\n");  
}  
  
system("PAUSE"); return 0;  
}
```

Esercizio 2

Scrivere un programma che richiede all'utente un operatore matematico + - / * oppure x (usare una variabile di tipo char) e stampare a video il nome dell'operazione o un messaggio di errore se operatore non riconosciuto:

+ → somma

- → sottrazione

: → divisione

/ → divisione

* → prodotto

x → prodotto (x: carattere x minuscolo)

Creare 2 soluzioni nello stesso programma:

- una con l'utilizzo del costrutto if/if-else
- una con l'utilizzo del costrutto switch

Esercizio 2 – Soluzione

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    char op;
    printf("inserisci operatore: "); fflush(stdin); scanf("%c",&op);
    /* opzione 1 prima versione */
    if (op=='+') {
        printf("somma\n");
    } else {
        if (op=='-') {
            printf("sottrazione\n");
        } else {
            if (op=='*' || op=='x') {
                printf("prodotto\n");
            } else {
                if (op=='/') {
                    printf("divisione\n");
                } else {
                    printf("Errore!!\n");
                }
            }
        }
    }
}
```


Esercizio 2 – Soluzione

```
/* opzione 1 seconda versione - non ottimizzata!*/  
if (op=='+') {  
    printf("somma\n");  
}  
if (op=='-') {  
    printf("sottrazione\n");  
}  
if (op=='*' || op=='x') {  
    printf("prodotto\n");  
}  
if (op=='/' || op==':') {  
    printf("divisione\n");  
}  
if (op!='+' && op!='-' && op!='*' && op!='x' && op!='/') {  
    printf("Errore!!\n");  
}
```

Esercizio 2 – Soluzione

```
/* soluzione 3*/
```

```
switch(op) {
```

```
    case '+':    printf("somma\n"); break;
```

```
    case '-':    printf("sottrazione\n"); break;
```

```
    case '*':
```

```
    case 'x':    printf("prodotto\n"); break;
```

```
    case ':':
```

```
    case '/':    printf("divisione\n"); break;
```

```
    default:     printf("Errore!!\n");
```

```
}
```

```
return 0;
```

```
}
```

Esercizio 3

Scrivere un programma che

- richiede all'utente un mese (in formato numerico)
- stampa a video il mese in formato letterale

Creare 2 soluzioni nello stesso programma:

- una con l'utilizzo del costrutto if/if-else
- una con l'utilizzo del costrutto switch

Esempio:

Input: 1 → Output: gennaio

Input: 3 → Output: marzo

Esercizio 3 - Soluzione

```
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char *argv[]){
    int mese; printf("inserisci mese: "); scanf("%d",&mese);
    /* opzione 1 prima versione */
    if (mese==1){
        printf("gennaio\n");
    } else {
        if (mese==2){
            printf("febbraio\n");
        } else {
            if (mese==3){
                printf("marzo\n");
            } else {
                if (mese==4){
                    printf("aprile\n");
                } else {
                    // . . . Da completare
                }
            }
        }
    }
}
```

Esercizio 3 - Soluzione

```
/* opzione 1 seconda versione */  
if (mese==1) { printf("gennaio\n"); }  
if (mese==2) { printf("febbraio\n"); }  
if (mese==3) { printf("marzo\n"); }  
if (mese==4) { printf("aprile\n"); }  
if (mese==5) { printf("maggio\n"); }  
if (mese==6) { printf("giugno\n"); }  
//. . . Da completare
```

Esercizio 3 - Soluzione

```
/* soluzione 2*/
```

```
switch (mese) {  
    case 1:      printf("gennaio\n"); break;  
    case 2:      printf("febbraio\n"); break;  
    case 3:      printf("marzo\n"); break;  
    case 4:      printf("aprile\n"); break;  
    case 5:      printf("maggio\n"); break;  
    case 6:      printf("giugno\n"); break;  
    case 7:      printf("luglio\n"); break;  
    case 8:      printf("agosto\n"); break;  
    case 9:      printf("settembre\n"); break;  
    case 10:     printf("ottobre\n"); break;  
    case 11:     printf("novembre\n"); break;  
    case 12:     printf("dicembre\n"); break;  
    default:     printf("????\n");  
}
```

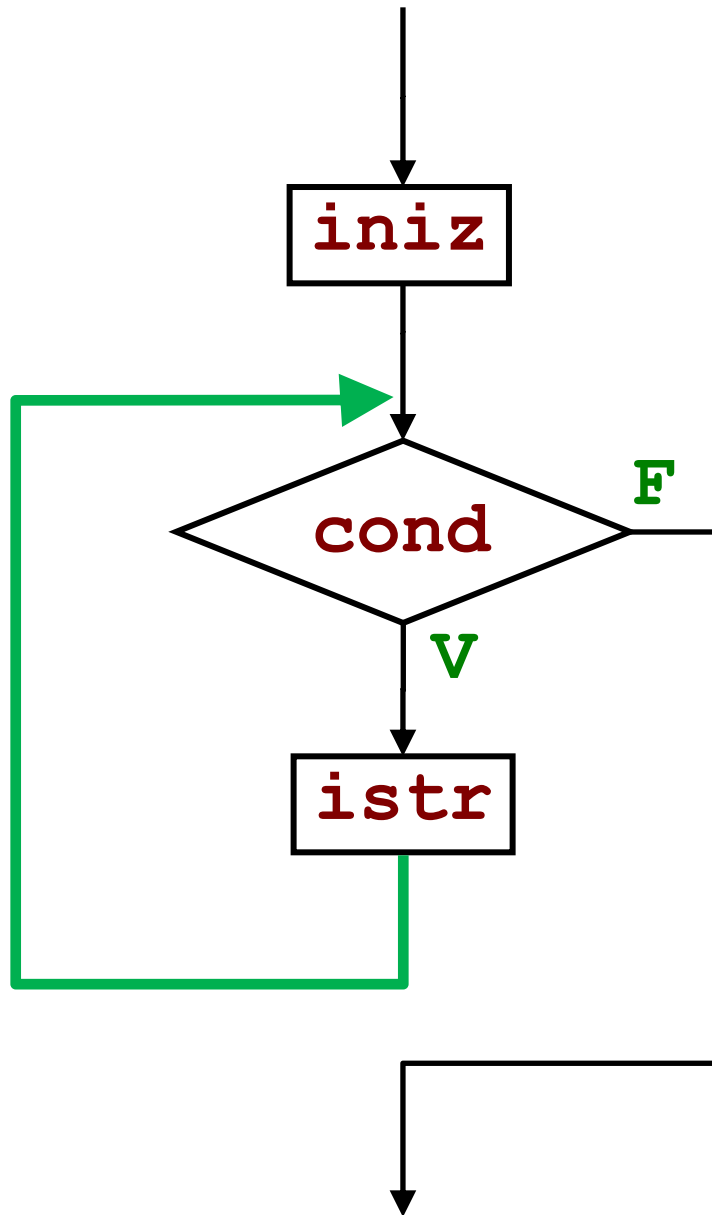
```
system("PAUSE");  
return 0;
```

```
}
```

Non necessario se si è sicuri che
la variabile mese contenga solo
valori tra 1 e 12

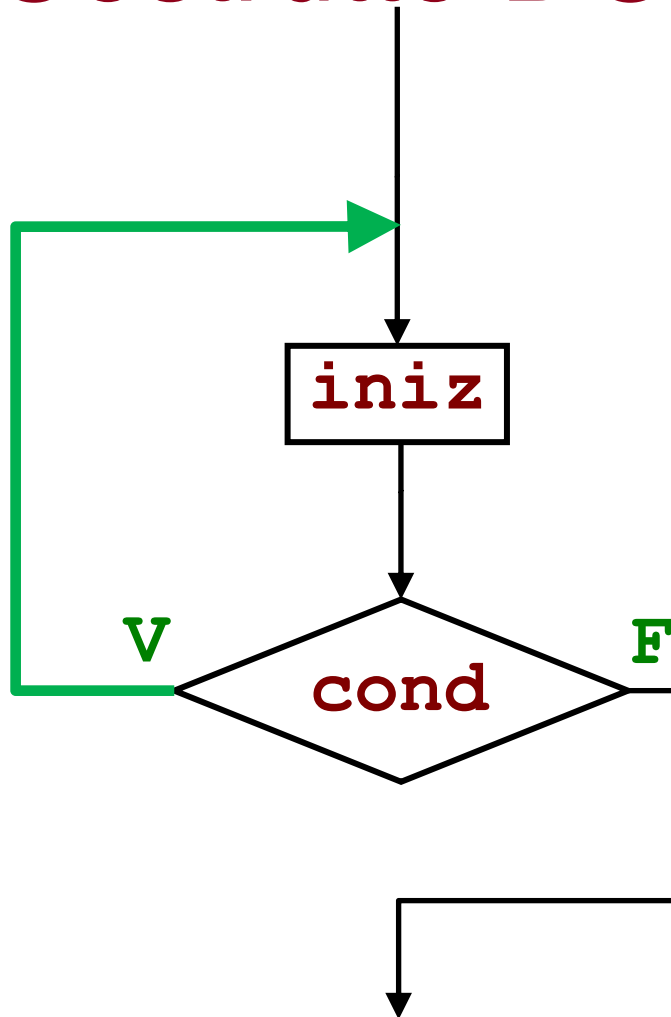
=> Usare controllo input

Costrutto WHILE



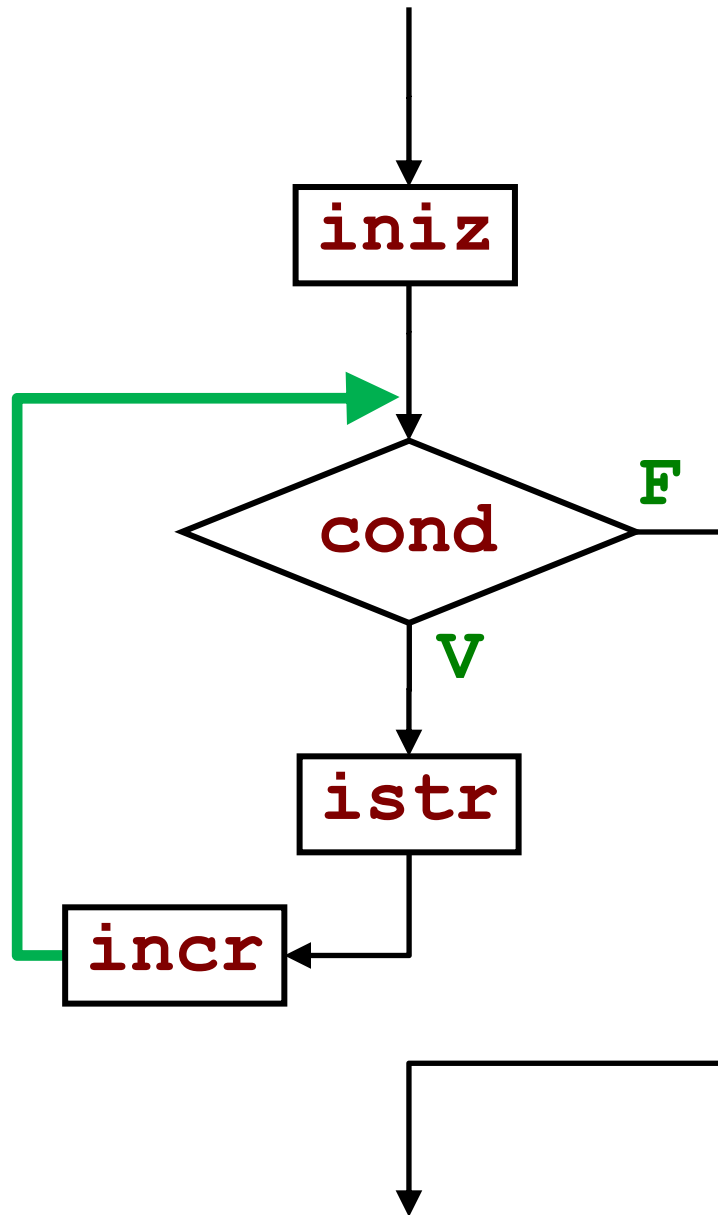
```
...  
while( cond ){  
    blocco  
    istruzioni  
}  
...
```


Costrutto DO-WHILE



```
...  
do {  
    blocco  
    istruzioni  
}while( cond );  
...
```

Costrutto FOR



```
...  
for(iniz ; cond ; incr){  
    blocco  
    istruzioni  
}  
...
```

Esercizio 4 controllo input

Modificare la soluzione dell'esercizio 3

Cambiando l'acquisizione del valore del mese (variabile mese) in modo che ci sia il controllo che il valore sia compreso tra 1 e 12

NOTA: usare un ciclo **do-while** per il controllo di validità per l'input dei dati che preveda al massimo 3 tentativi

Esempio 4 soluzione

```
int mese;
printf("inserisci mese:"); scanf("%d",&mese);
//con l'if faccio un solo controllo
if (mese<1 || mese>12) {
    printf("valore mese errato! ");
    printf("inserisci mese:"); scanf("%d",&mese);
}
```

```
int mese;
printf("inserisci mese:"); scanf("%d",&mese);
//fintantochè il valore è errato continuo a richiederlo
while (mese<1 || mese>12) {
    printf("valore mese errato! ");
    printf("inserisci mese:"); scanf("%d",&mese);
}
```

Esempio 4 soluzione

```
int mese, tentativi = 1;
printf("tentativo num %d inserisci mese:", tentativi);
scanf("%d", &mese);
//fintantochè il valore è errato continuo a richiederlo
//ma solo per 3 volte al massimo
while ( (mese<1 || mese>12) && tentativi<=3) {
    printf("valore errato! ");
    printf("tentativo num %d inserisci mese:", tentativi);
    scanf("%d", &mese);
    tentativi++;
}
if (mese>=1 && mese<=12) {
    switch(mese){
        //controllo valore mese
    }
} else {
    printf("spiacente numero tentativi esaurito\n");
}
```

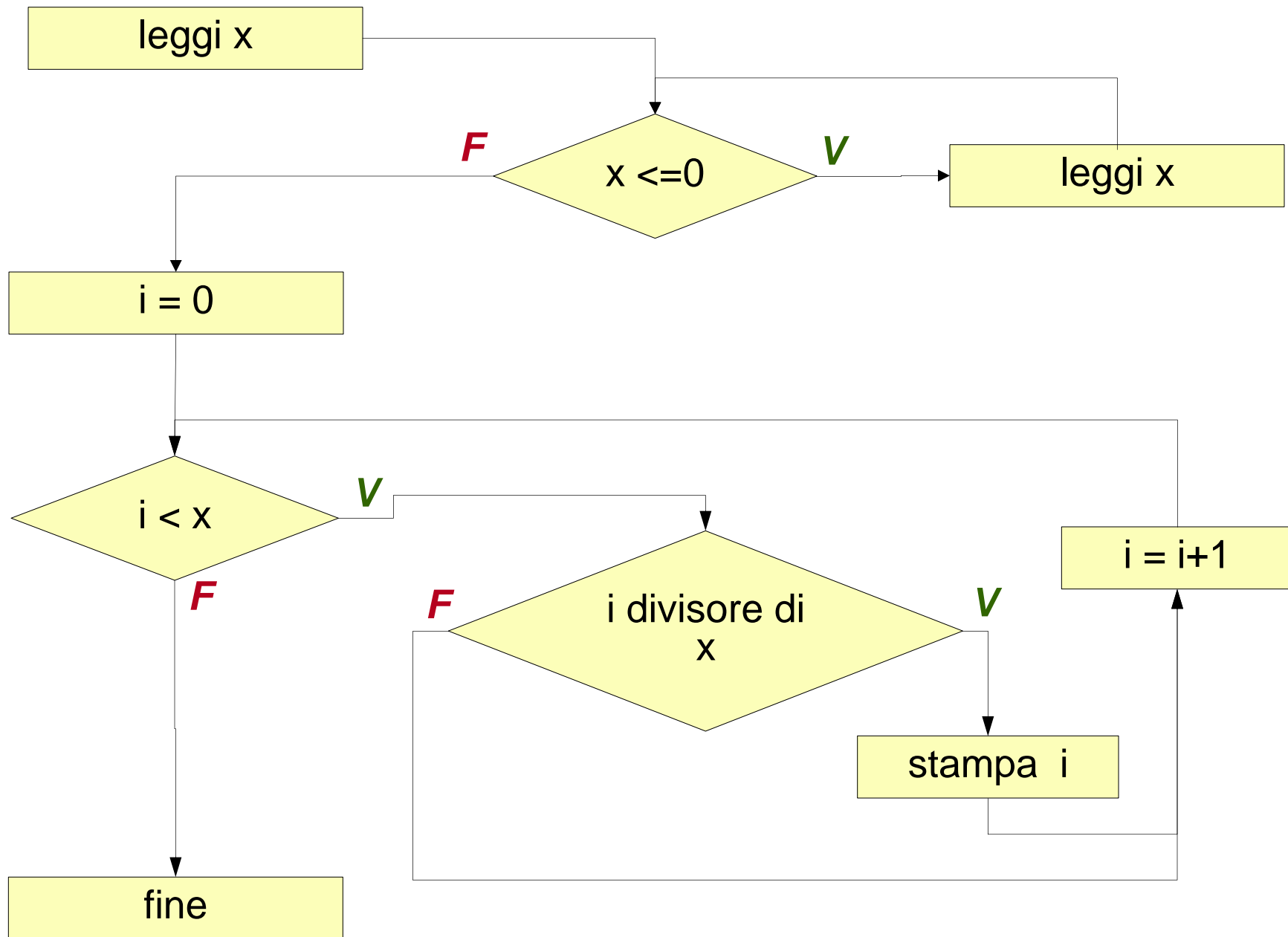
Esempio 4 soluzione do-while

```
int mese, tentativi = 1;
//visto che almeno 1 volta dove leggere il mese
//potrei usare il do-while
do{
    printf("tentativo num %d inserisci mese:",tentativi);
    scanf("%d",&mese);
    if (mese<1 || mese>12) {
        printf("valore errato! ");
    }
    tentativi++;
} while ( (mese<1 || mese>12) && tentativi<=3) ;
if (mese>=1 && mese<=12) {
    switch(mese){
        //controllo valore mese
    }
} else {
    printf("spiacente numero tentativi esaurito\n");
}
```

Esercizio 5

Scrivere un programma che richiedi un numero ***X*** all'utente, stampa tutti i divisori di ***X***

- Controllo dell'input (usare ciclo do-while)
 - il numero deve essere maggiore di 0
- n è divisore di x
 - se resto della divisione x/n è nullo
 - $x \% n == 0$
- **NOTA:** tra i divisori si escludono $n==0$ (non ha senso), $n==1$ e $n==x$ perchè divisori banali



Esercizio 5 - soluzione

```
int i, x;

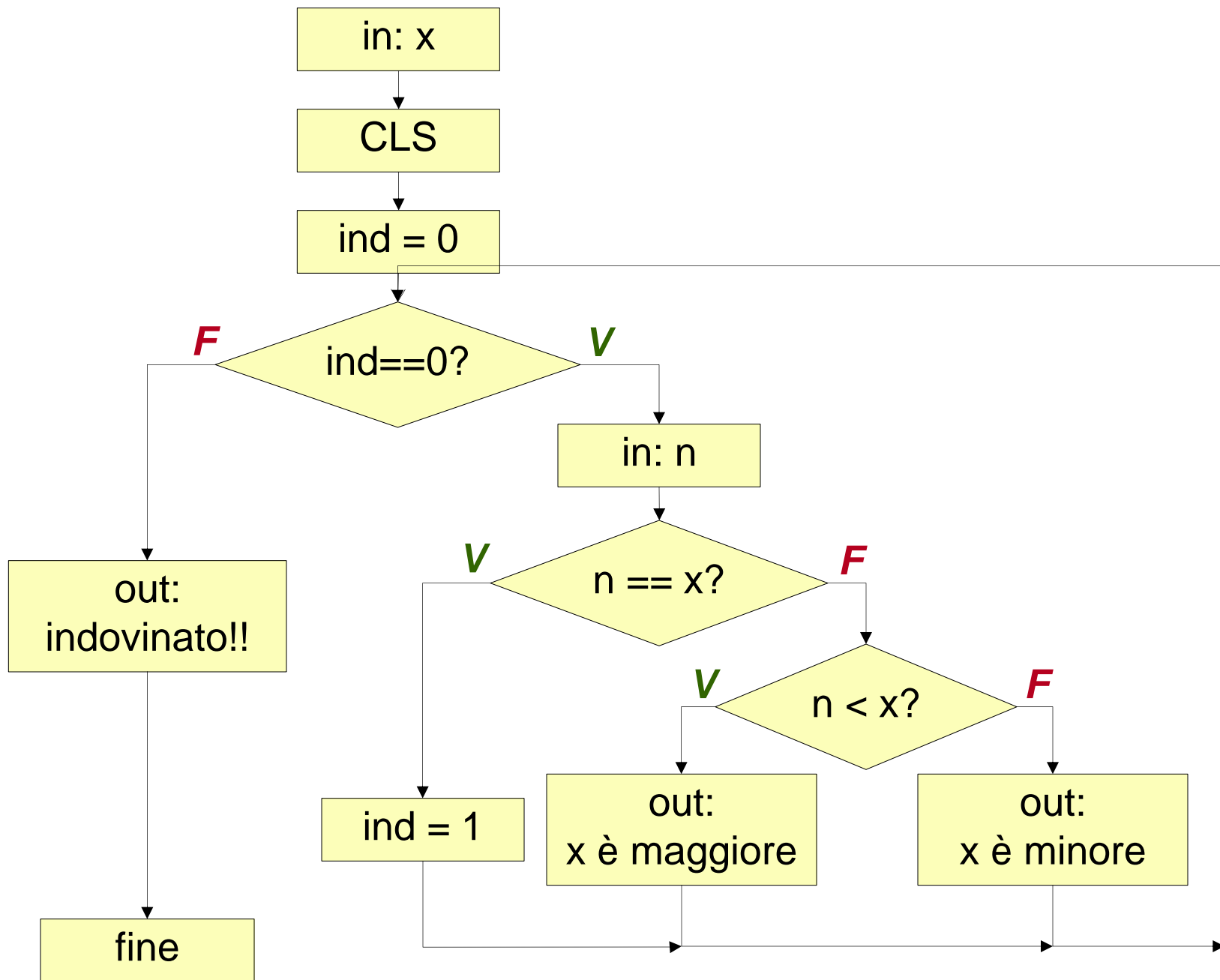
/* controllo input */
do{
    printf("Inserisci numero positivo: ");
    scanf("%d", &x);
}while( x<=0 );

/* ricerca divisori */
/* i=0 non ha senso, i=1 e i=x sono sempre divisori */
i = 2;
/* while (i < x/2) : riduce il numero di cicli */
while(i < x){
    if( x%i == 0 ){
        printf("Divisore: %d\n", i);
    }
    i++;
}

system("PAUSE");
```

Esercizio 6

- Riesci ad indovinare il numero che ho inserito? Dopo aver inserito un valore numerico da tastiera, l'altro giocatore deve indovinarlo. Alcuni aiuti sono forniti (es. maggiore/minore).
- **Da sapere/ricordare:**
 - usate `system("CLS")` ; per cancellare il testo mostrato sulla console... altrimenti è troppo facile!
- **In aggiunta:**
 - limitare il numero inserito da tastiera con un massimo. Cosa succede se il primo giocatore inserisce un numero maggiore?
 - **limitare il numero di tentativi a 3**
 - **Dire se il numero inserito è maggiore o minore del numero da indovinare**
 - come fare a giocare più volte?
 - modificare il programma per giocare con i caratteri



Esercizio 6 - soluzione

```
int main(){
    int valore, tentativo, indovinato;
    printf("Inserisci il valore da indovinare\n");
    scanf("%d", &valore);
    system("CLS");
    indovinato = 0;
    while (!indovinato) { /* indovinato==0 */
        printf("Indovina il numero\n");
        scanf("%d", &tentativo);
        if (tentativo == valore) {
            indovinato = 1;
        } else {
            if (tentativo > valore) {
                printf("Il numero da indovinare e' piu' piccolo\n");
            } else {
                printf("Il numero da indovinare e' piu' grande\n");
            }
        }
    }
    printf("Non ci posso credere ;-) hai indovinato!\n");
    system("PAUSE");
}
```

Esercizio 6 – soluzione ver2

```
int main(){
    int valore, tentativo, indovinato;
    printf("Inserisci il valore da indovinare\n");
    scanf("%d", &valore);
    system("CLS");
    indovinato = 0;
    while (!indovinato) {
        printf("Indovina il numero\n");
        scanf("%d", &tentativo);
        if (tentativo == valore) {
            printf("Non ci posso credere ;-) hai indovinato!\n");
            indovinato = 1;
        } else {
            if (tentativo > valore){
                printf("Il numero da indovinare e' piu' piccolo\n");
            } else {
                printf("Il numero da indovinare e' piu' grande\n");
            }
        }
    }
    system("PAUSE");
}
```

Esercizio 6 – soluzione estesa

```
int val, tentativo, indovinato, n_tentativo,
char continua='s';
while(continua=='s'){
    printf("Numero da indovinare\n");  scanf("%d", &val);
    system("CLS");
    indovinato = 0; n_tentativo = 0;
    while( !indovinato && n_tentativo<8 ) {
        printf("Indovina il numero\n");
        scanf("%d", &tentativo);
        if (tentativo == val) {
            printf("Non ci posso credere ;-) hai indovinato!\n");
            indovinato = 1;
        } else {
            if (tentativo > valore){
                printf("Il numero da indovinare e' piu' piccolo\n");
            } else {
                printf("Il numero da indovinare e' piu' grande\n");
            }
        }
        n_tentativo++;
    }
    if( !indovinato ) { printf("Il numero era: %d\n", val);}
    printf("continui? s/n"); fflush(stdin); scanf("%c", &continua);
}
```


Esercizio 7

Stampare a video la tavola delle moltiplicazioni. Gli estremi sono inseriti da tastiera.

es (estremi 2 e 6)

1	2	3	4	5	6
2	4	6	8	10	12

Da sapere/ricordare:

quando usate la `printf` potete indicare il numero di spazi che si vuole riservare per il valore della variabile: es. `printf("%3d")` riserva 3 spazi per il valore della variabile anche se non sono necessari.

Soluzione

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

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

```
int main() {
```

```
    int x, y, limiteX, limiteY, valore;
```

```
    printf("Inserire il primo limite: ");    scanf("%d",&limiteX);
```

```
    printf("Inserire il secondo limite: "); scanf("%d",&limiteY);
```

```
    x = 1;
```

```
    while (x <= limiteX) {
```

```
        y = 1;
```

```
        while (y <= limiteY) {
```

```
            valore = x * y;
```

```
            printf("%3d", valore);
```

```
            y = y + 1; //oppure y++;
```

```
        }
```

```
        printf("\n");
```

```
        x = x + 1;    //oppure x+=1;
```

```
    }
```

```
    system("PAUSE");
```

```
}
```