
Le Strutture di controllo Del Linguaggio C

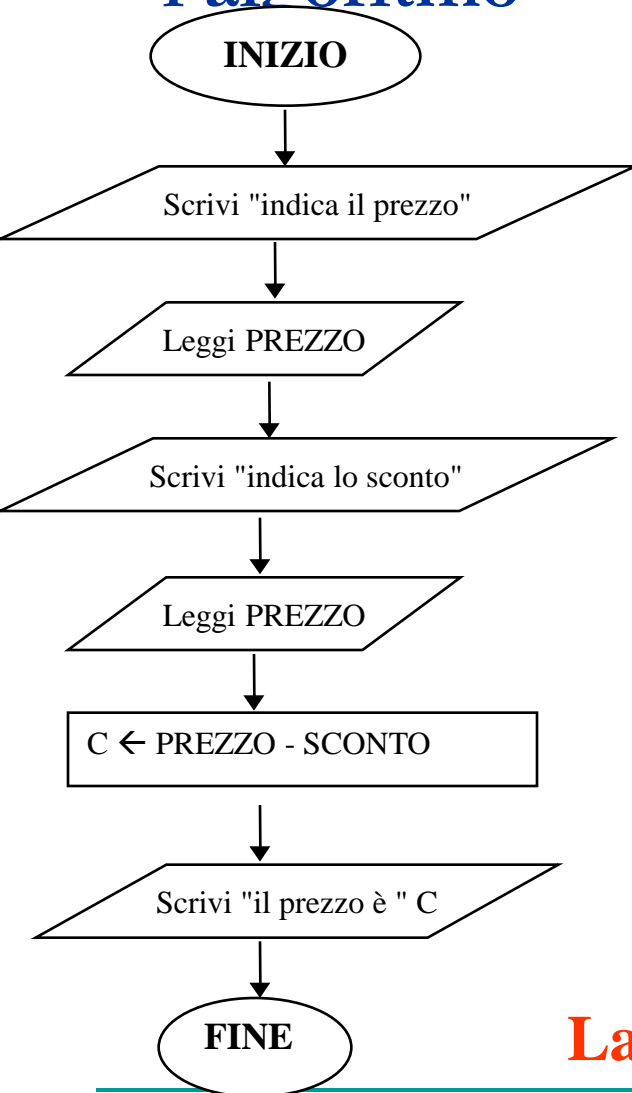
**Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni**

STRUTTURE DI CONTROLLO

- PRIMITIVE
 - SEQUENZA
 - SELEZIONE (o scelta logica)
 - ITERAZIONE

- NON PRIMITIVE
derivate da quelle primitive
 - ITERAZIONE PREDEFINITA
 - SCELTA (o selezione) MULTIPLA

Alcune osservazioni sulle tecniche per descrivere l'algoritmo



Questi primi algoritmi sono abbastanza semplici. Prima di vederne di più complessi è opportuno fare una semplificazione:

Assumeremo di non indicare le istruzioni di output che servono per chiedere all'utente di indicare i dati. Ad esempio:

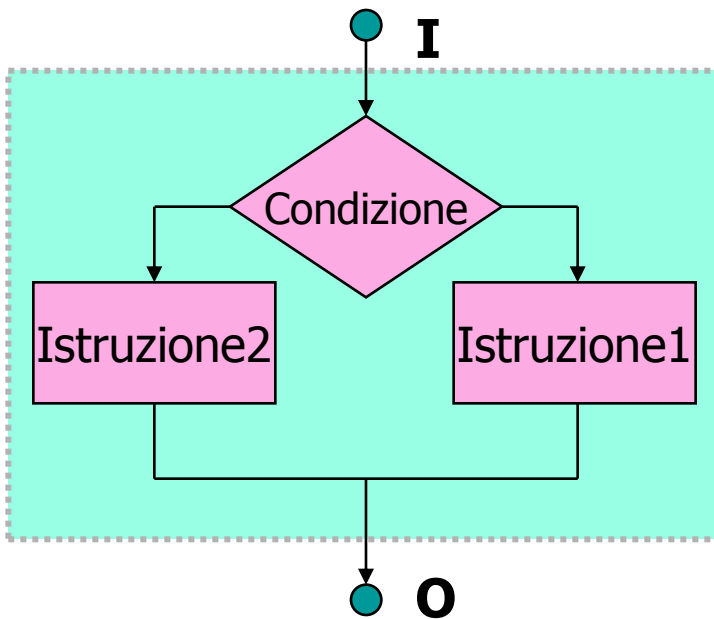
verrà sostituita dalla sola istruzione



La tabella dei dati deve essere sempre descritta

LA SELEZIONE (scelta logica)

Con questa struttura si seleziona *in modo esclusivo* una determinata sequenza/blocco di istruzioni da eseguire sulla base di una <condizione> rappresentata da una espressione booleana , che può essere *vera o falsa*



Se (condizione)

```
{  
    blocco_istruzioni_1  
}
```

Altrimenti

```
{  
    blocco_istruzioni_2  
}
```

Sintassi della struttura di Selezione

In C la struttura di selezione si codifica

così

```
if (condizione)
    istruzione1;
else
    istruzione2 ;
```

Se sono presenti più istruzioni in ciascun ramo

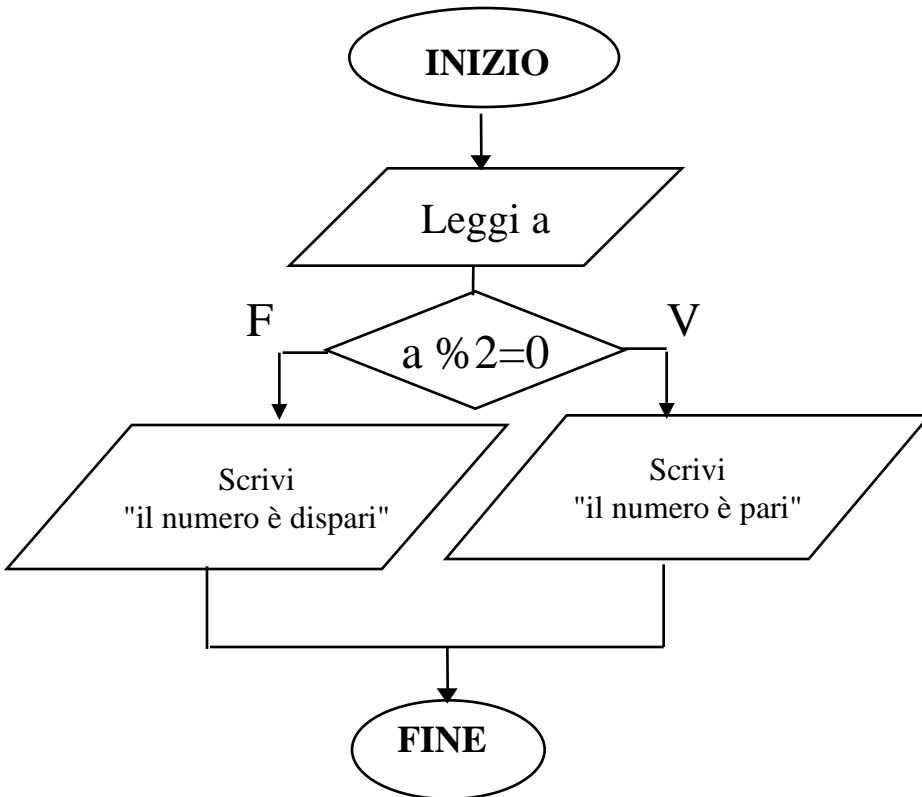
```
if (condizione)
{
    istruzione1;
    istruzione2;
}
else
{
    istruzione1;
    istruzione2;
}
```

RICORDA:

Le istruzioni che fanno parte della sequenza sono racchiuse tra le parentesi graffe.

Esempio Istruzione di Selezione a due vie

PROBLEMA: dato un numero si comunichi se il numero è pari o dispari



```
#include <stdio.h>
#include <conio.h>
int main()
{
    int a;
    scanf("%d",& a); // leggi numero da terminale
    if ((a % 2) == 0)
    {
        printf("il numero e' pari");
    }
    else
    {
        printf("il numero e' dispari");
    }
    getch();
}
```

istruzione precedente

istruzione if-else

istruzione successiva

LA SELEZIONE A UNA VIA

In alcuni casi è possibile scegliere di **compiere** o **non compiere** un'azione;

in tal caso l'azione da eseguire **deve essere** posta sul ramo del **SI**.

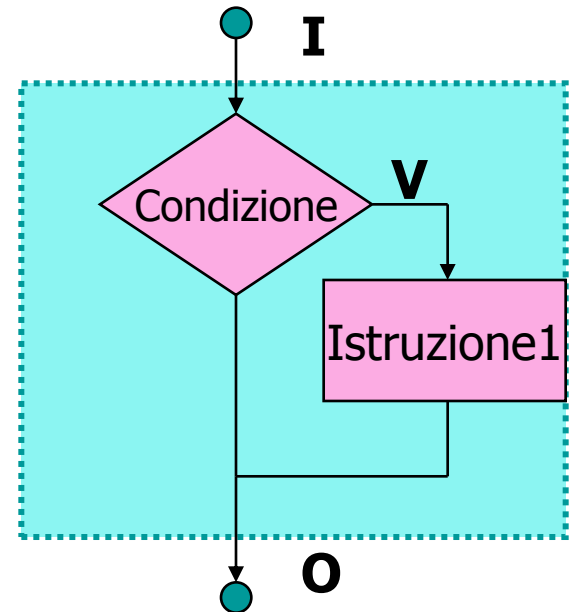
In questo caso la codifica in C++ è:

```
if (condizione)  
    istruzione1;
```

o nel caso di più istruzioni:

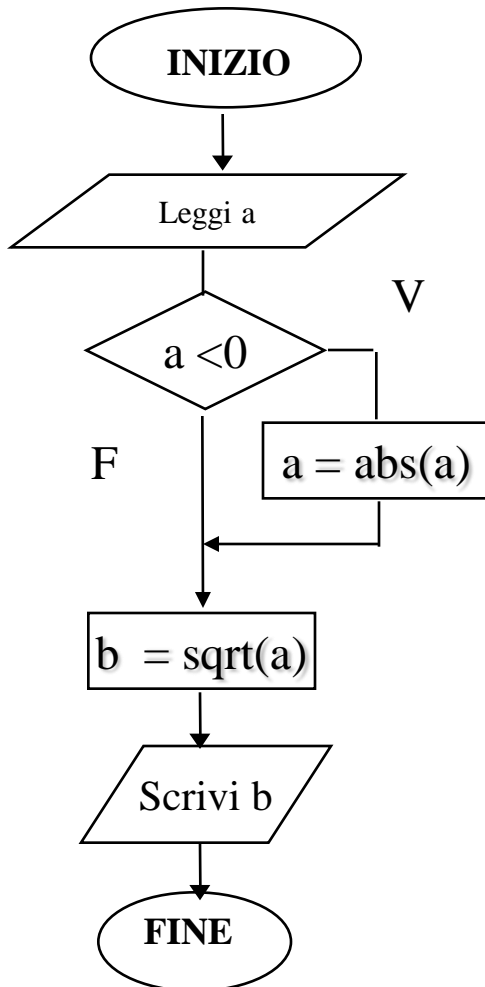
```
if (condizione)  
{  
    istruzione1;  
    istruzione2;  
    .....;  
}
```

La clausola **else** (altrimenti) in queste situazioni non viene codificata



Esempio Istruzione di Selezione a una via

PROBLEMA: dato un numero intero si comunichi il valore della radice quadrata del suo valore assoluto



```
#include <stdio.h>
#include <conio.h>
#include <math.h>
int main()
{
    int a; float b;
    scanf("%d",& a); // leggi numero da terminale
    if (a < 0)
    {
        a = abs(a); // calcola valore assoluto di a
    }
    b = sqrt(a);
    printf( "Radice quadrata: ", b);
    getch();
}
```

istruzione precedente

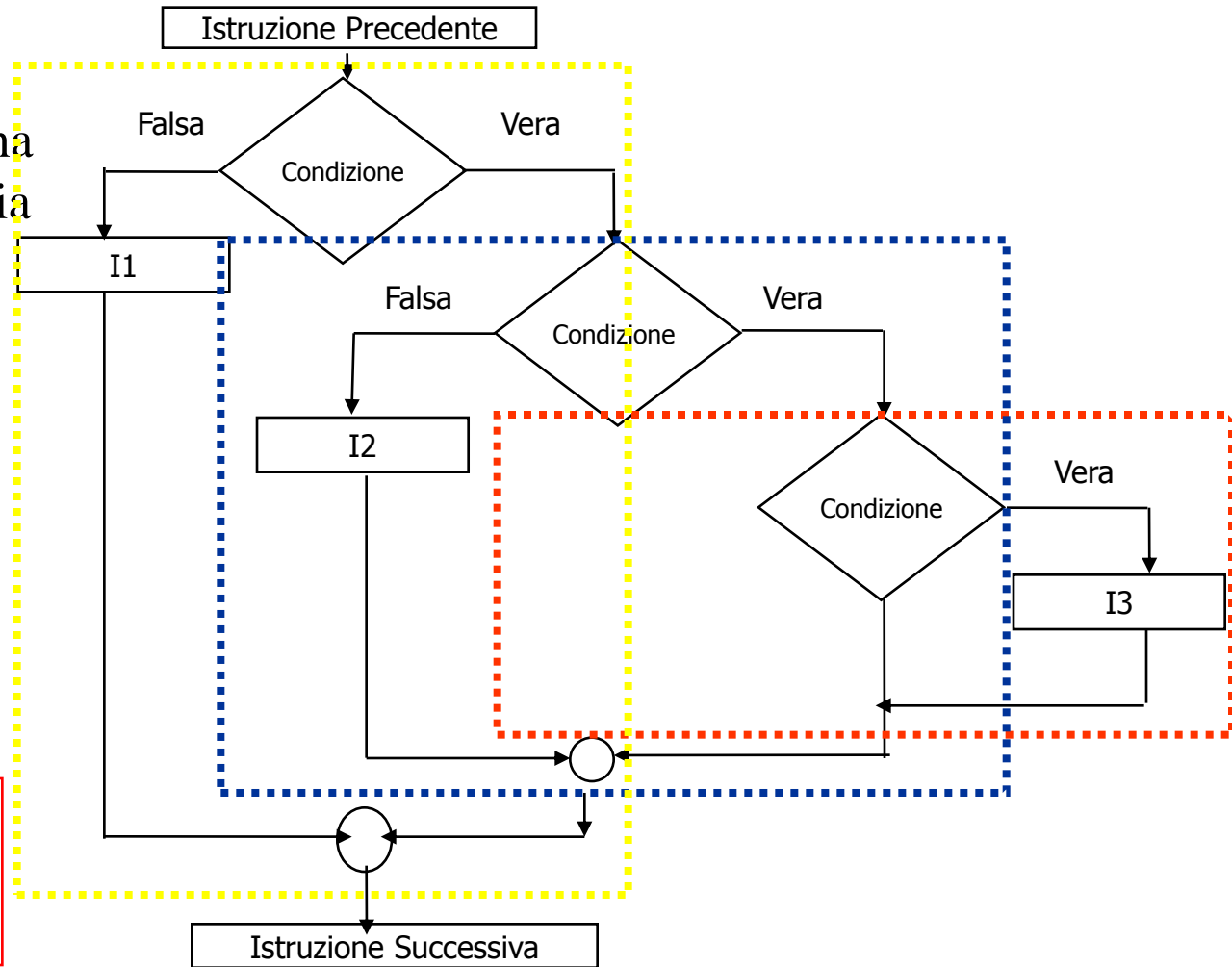
istruzione if

istruzioni successive

Strutture di Controllo if-else Nidificati

A volte si possono usare istruzioni di selezione (a una o a due vie) **nidificate**, ossia le istruzioni da eseguire al verificarsi della condizione sono a loro volta istruzioni condizionali

Occorre ricordare che ogni blocco deve avere un inizio e una fine



ITERAZIONE CON CONTROLLO IN CODA (do...while)

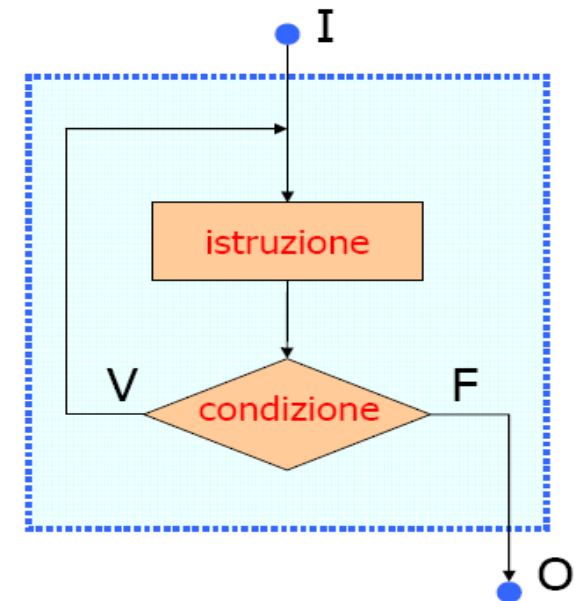
Per **iterazione** si intende la **ripetizione** di **una o più azioni** sotto il **controllo di una condizione**.

Il gruppo di azioni da ripetere è detto **corpo del ciclo**

```
Do{  
    .....  
    Istruzioni  
    .....  
}While(condizione);
```

E' necessario che all'interno del corpo vi sia una istruzione che modifichi il valore della condizione, altrimenti il ciclo "va in loop"

Diagramma a blocchi



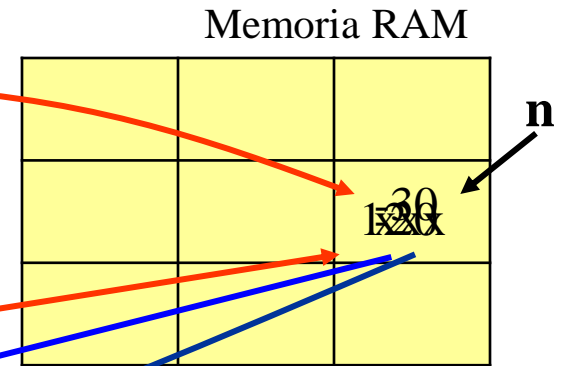
Esecuzione di un ciclo do..while

Questa struttura può essere utilizzata per il "*controllo dei dati in input*"

ESERCIZIO: Leggi ripetutamente un numero finché il numero letto è > 0

```
→ int n;  
→ do  
→ {printf( "Dammi un numero.");  
→ scanf("%d",& n);  
→ if (n<=0)  
→ printf("numero non positivo- ripetere");  
→ } //chiusura del corpo del ciclo  
→ while (n<=0);  
→ printf("Fine");
```

La struttura do..while, consente di eseguire ripetutamente il corpo del ciclo anche per un numero di volte indeterminato



Esempio di iterazione con controllo in coda

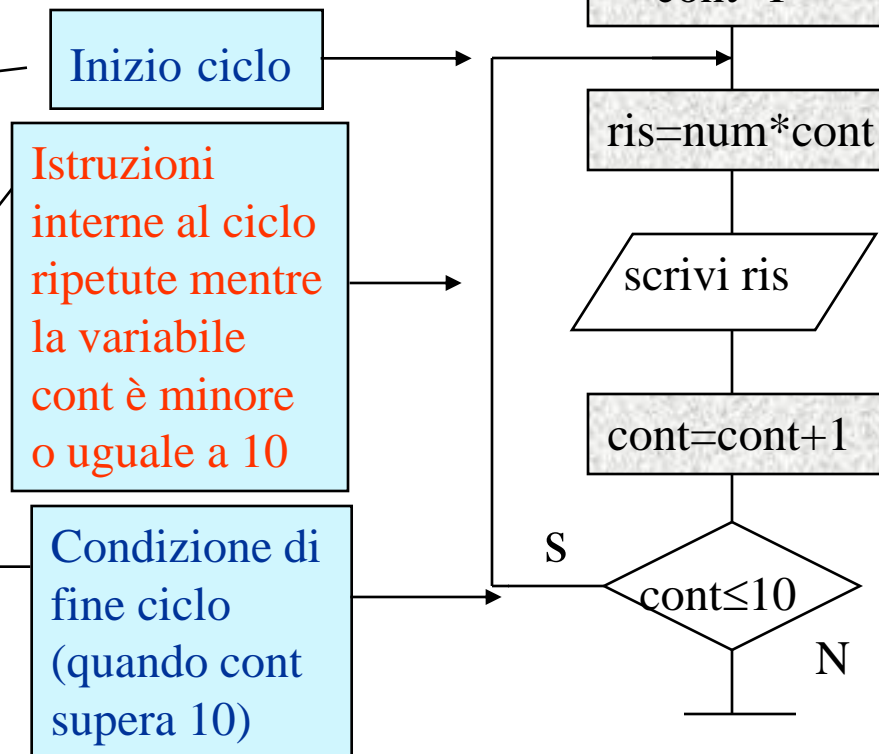
ESERCIZIO: Visualizzazione dei primi 10 multipli di 5

Codifica in C:

```
num=5; cont=1;
```

```
do  
{  
    ris=num*cont;  
    printf(“%d“,ris);  
    cont++;  
}
```

```
while (cont<=10);
```



RICORDA:

Quando si fa riferimento ad una condizione (sia nella scelta che nell'iterazione) si intende che è possibile utilizzare anche **condizioni composte**.

	In C
AND prodotto logico	&&
OR somma logica	
NOT negazione	!

Si ottengono condizioni composte legando due o più condizioni con gli operatori logici:

Esempi in C: **OR** nella selezione

```
if ((a>0) || (b>0)) .....
```

AND nella ripetizione

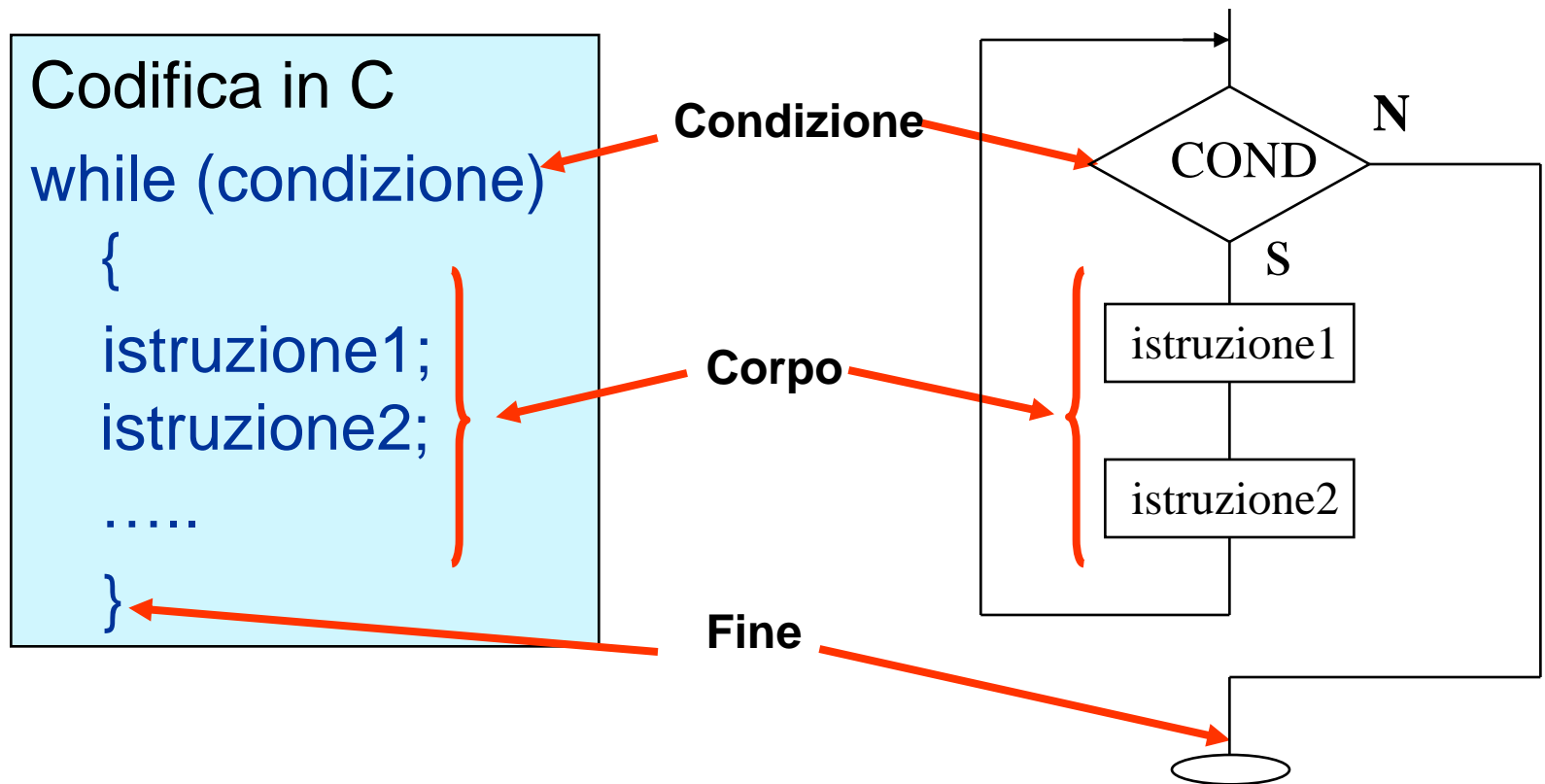
```
do { .....
```

```
.....
```

```
} while ((a>0) && (b>0));
```

ITERAZIONE CON CONTROLLO IN TESTA (while ...)

Il while verifica la condizione all'inizio del ciclo per cui se inizialmente la condizione risulta falsa il corpo del ciclo non viene mai eseguito

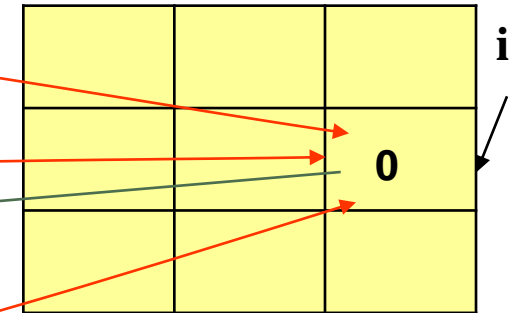


Esecuzione di un ciclo while

ESERCIZIO: Leggi ripetutamente un numero in ingresso e calcolane la radice quadrata fino a quando il numero letto è > 0

```
int i;  
Printf( "Dammi il primo num.");  
scanf( "%d",&i);  
while ( i > 0 )  
{ printf("%d",sqrt(i));  
  printf("Dammi un altro num.");  
  scanf( "%d",&i);  
}  
Printf( "Fine");
```

Memoria RAM



Anche la struttura while..., consente di eseguire ripetutamente il corpo del ciclo anche per un numero di volte indeterminato

Esempio di iterazione con controllo in testa

ESERCIZIO: Visualizzazione dei primi 10 multipli di 5

Codifica in C:

```
num=5; cont=1;
```

```
while (cont<=10)
```

```
{
```

```
    ris=num*cont;
```

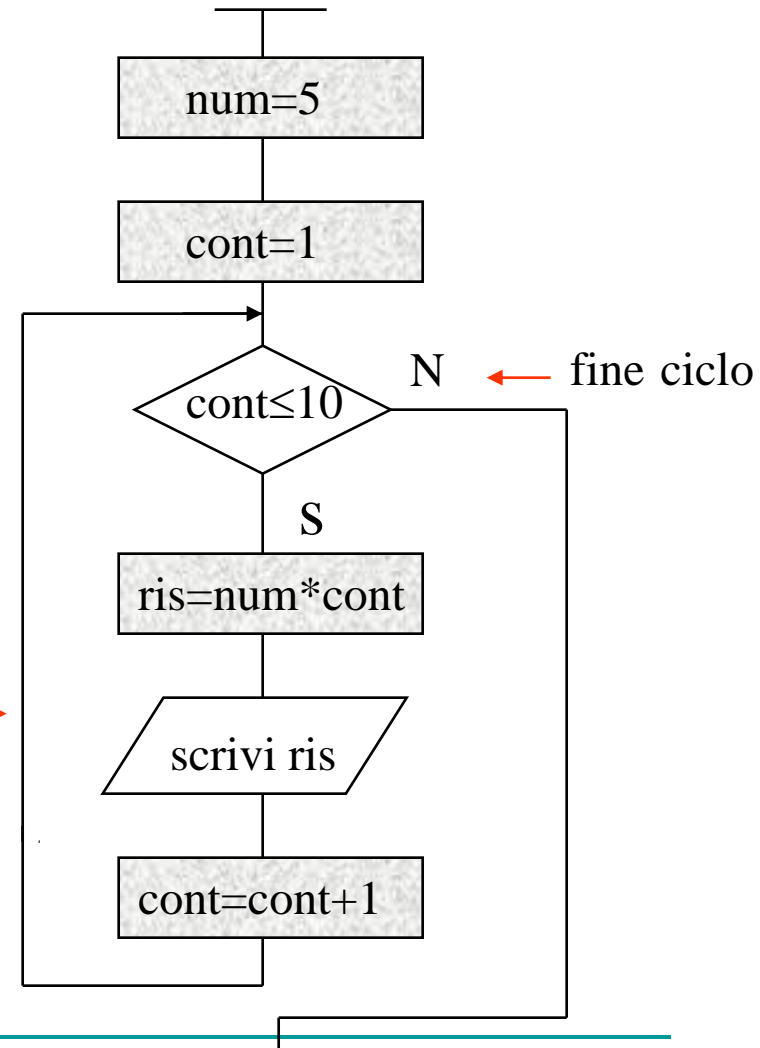
```
    printf("%d ",ris);
```

```
    cont++;
```

```
}
```

Condizione di
inizio ciclo
(mentre $cont \leq 10$)

Istruzioni ripetute
all'interno del ciclo



ITERAZIONE PREDEFINITA (for)

Ha questo nome poiché si tratta di un'iterazione in cui si conosce **quante volte il ciclo viene ripetuto**.

L'iterazione enumerativa è derivata dalla iterazione con controllo in testa, perciò la sua rappresentazione con il flow-chart sarà simile a quella dell'iterazione con controllo in testa.

E' però presente una **variabile intera di controllo** del numero ripetizioni del corpo del ciclo.

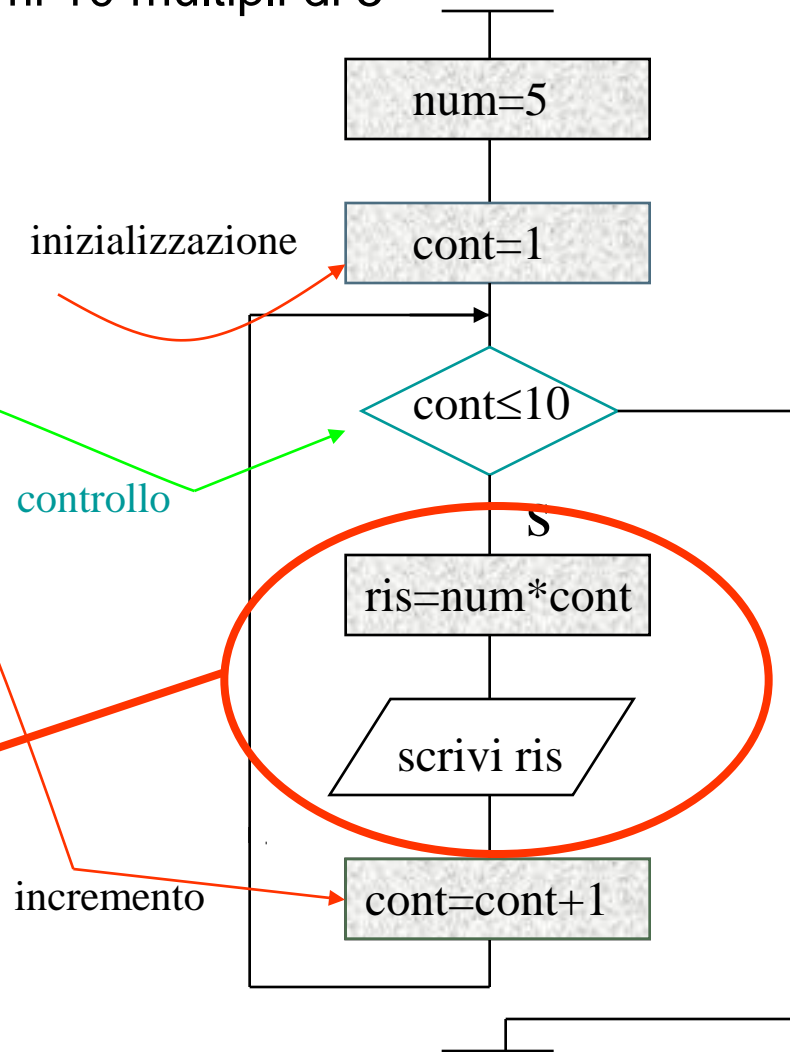
Tale variabile viene **inizializzata fuori ciclo, incrementata nel ciclo e controllata nella condizione di uscita dal ciclo (ciclo contatore)**.

Esempio di iterazione Predefinita

ESERCIZIO: Visualizzazione dei primi 10 multipli di 5

```
Codifica in C:  
  
num=5;  
for (cont=1;cont<=10;cont++)  
{  
    ris=num*cont;  
    printf(" %d",ris);  
}
```

Corpo del Ciclo



Codifica in C dell' iterazione Predefinita

L'istruzione for è composta da tre espressioni separate dal ';' la prima è di **inizializzazione**, la seconda di **controllo**, la terza di **incremento**.



```
for ( espressione1; espressione2 ; espressione3 )  
    istruzione;
```

L'ordine di esecuzione è il seguente:

- 1) esecuzione dell'**INIZIALIZZAZIONE** espressione1 (una sola volta)
- 2) controllo della **CONDIZIONE** espressione2 (ripetuta dopo ogni incremento)
- 3) esecuzione del **CICLO** istruzione (se la condizione è vera)
- 4) esecuzione dell'**INCREMENTO** espressione3 (ritorno al punto 2)

Quando si equivalgono while / for

Esempio di equivalenza strutture iterative while e for per iterazioni controllate da contatore in C:

```
float lato, area;
int i;
for (i=0; i<3; i++) {
    printf("Lato ? " );
    scanf("%d",&lato);
    area=lato*lato;
    printf( "Area: %f",area);
}
printf( "Fine");
```

```
float lato, area;
int i;
i=0;
while (i<3) {
    printf("Lato ? " );

    scanf("%f",&lato);
    area=lato*lato;

    printf( "Area: %f",area);

    i++;
}
printf( "Fine");
```

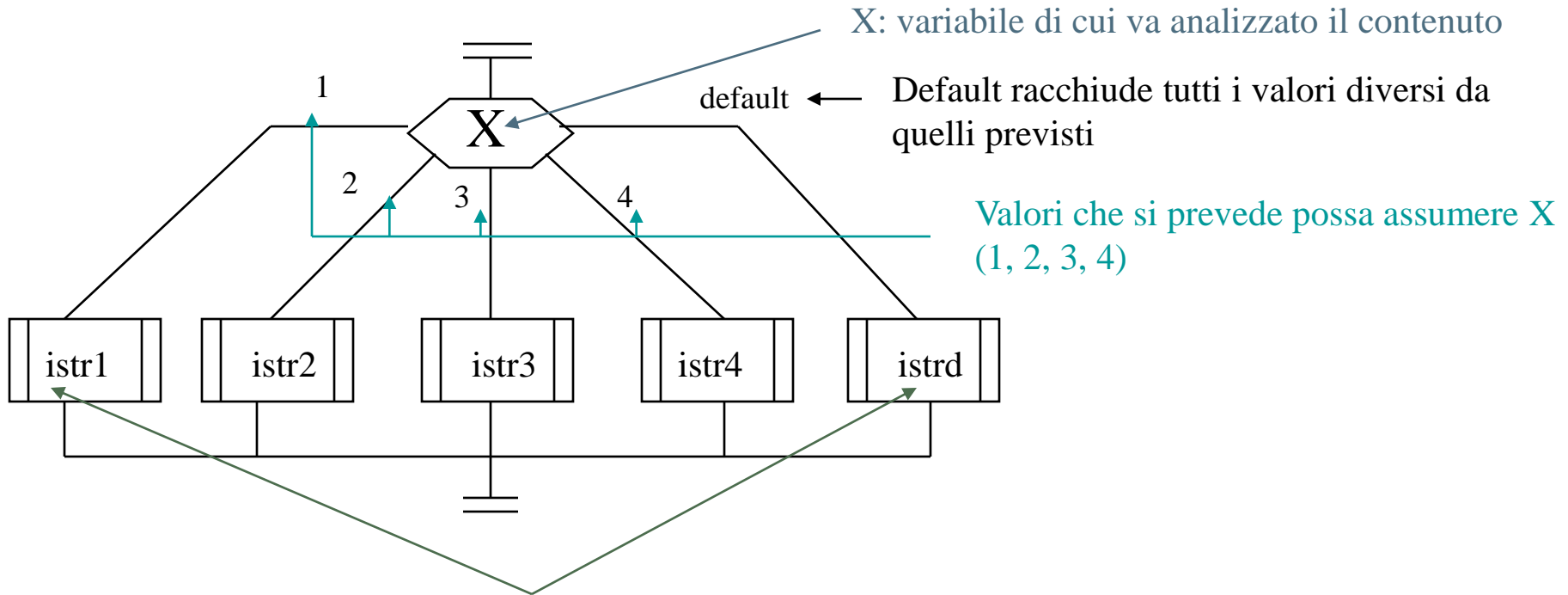
La scelta multipla

Esempio tipico di scelta multipla è quello di un programma che permette all'utente, attraverso un menù iniziale di scegliere tra le funzioni offerte.



- Tale situazione è risolvibile mediante una serie di scelte logiche
- Molti linguaggi di programmazione, tra cui il C, agevolano il programmatore mettendo a disposizione la sceita multipla

Rappresentazione della scelta multipla nel diagramma a blocchi



istr1, *istr2*, ... rappresentano le istruzioni da eseguire.

Se nella variabile X c'è il valore 1 viene eseguito il gruppo di istruzioni *istr1*, se in X c'è il valore 2 le istruzioni *istr2* e così via.

Sintassi della selezione multipla in C++

in C la **Selezione Multipla** è rappresentata dalla istruzione **switch-case**

```
switch (variabile){  
    case <costante1>: <sequenza di istruzioni1>;    break;  
    case <costante2>: <sequenza di istruzioni2>;    break;  
    ....  
    case <costanten>: <sequenza di istruzionin>;    break;  
    default : <sequenza di istruzioni>;            break;  
}
```

- le espressioni costanti **costante1**, **costante2**, ..., **costanten** devono essere di **tipo intero**
- In C i **caratteri sono assimilati agli interi** (ne viene considerata la codifica interna in ASCII)
- l'istruzione **break** serve per terminare lo **switch** dopo aver eseguito solo la sequenza associata ad una particolare scelta (**case**) e proseguire con l'istruzione successiva allo switch-case;
- nel costrutto può comparire la clausola **default** per individuare la sequenza di operazioni da compiere quando il valore esaminato non coincide con alcuno di quelli specificati

Semplice esempio di switch-case in C

ESERCIZIO: dato un numero intero, visualizza il nome del giorno della settimana corrispondente

```
main()
{
int num;
Printf("Dammi un giorno ( come numero): ");
scanf(" %d", &num) ;
    switch (num)
    {
        case 1: printf(" Lunedì" ); break;
        case 2: printf(" Martedì" ); break;
        case 3: printf("Mercoledì" ); break;
        case 4: printf("Giovedì" ); break;
        case 5: printf("Venerdì"); break;
        case 6: printf("Sabato"); break;
        case 7: printf("Domenica") ; break;
        default : printf("Giorno Inesistente" ); break;
    }
}
```