

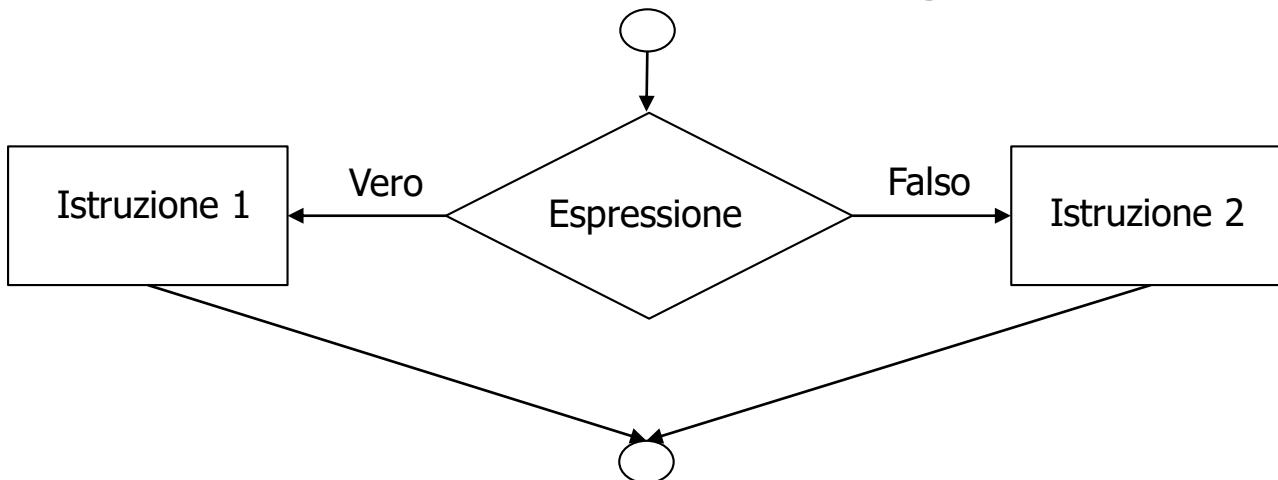
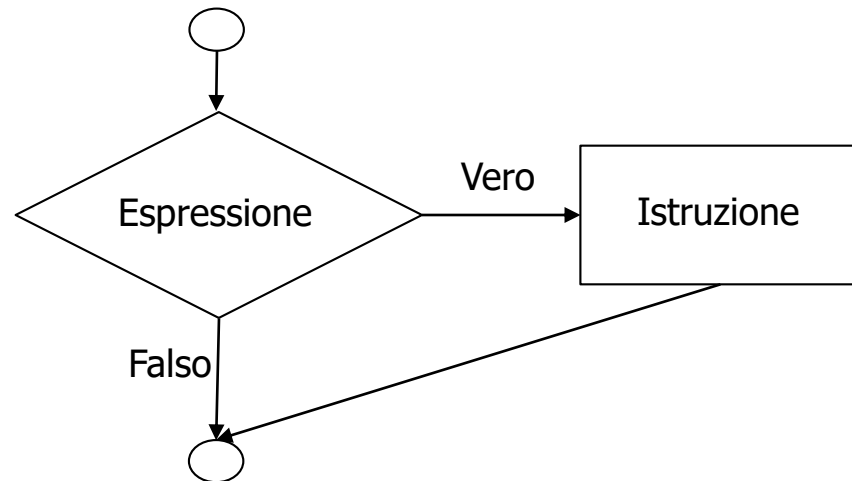


Istruzioni Condizionali in C

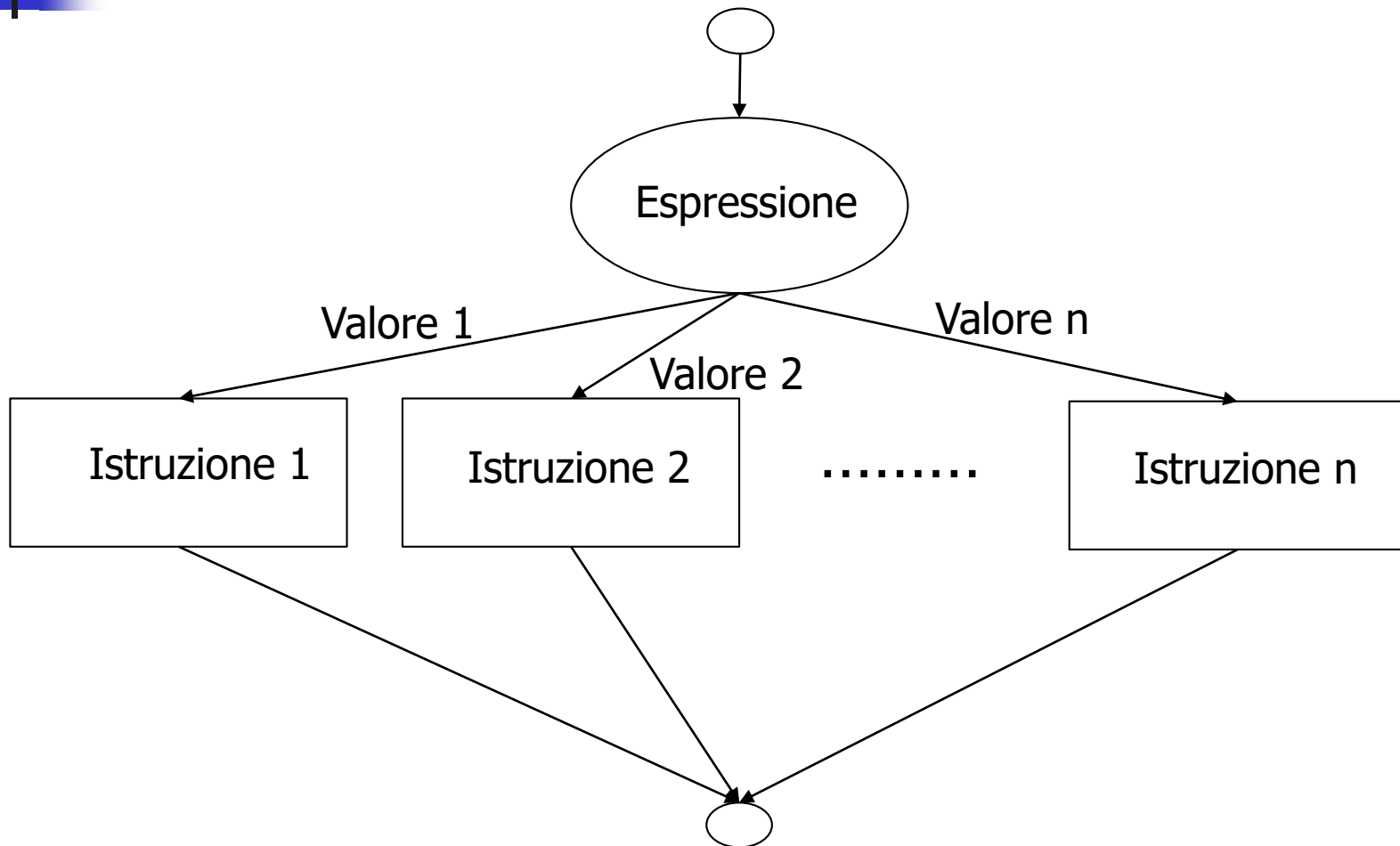
Concetti chiave

- Tipi di Istruzioni Condizionali
- Espressioni
- Blocchi d'istruzioni
- Istruzioni Condizionali: `if-else`
- Istruzioni Condizionali: `switch-case`
- Istruzione `break`

Istruzioni Condizionali



Istruzioni Condizionali





Espressioni

- Espressione: combinazione di variabili ed operatori
 - Operatori: Aritmetici, **Relazionali e Logici**, Assegnazione, Incremento/Decremento unitario, Condizionali e Virgola
- Valore Assunto: booleano (vero, falso)
- In linguaggio C:
 - Falso=0
 - Vero=qualunque valore diverso da zero
 - **x**, vero se x è diverso da zero, altrimenti è falsa



Espressioni

❖ Operatori Relazionali:

➤ Operatore di uguaglianza $==$

- è diverso da quello di assegnamento $=$

➤ Operatore di disequaglianza $!=$

➤ Operatori: $>$, $>=$, $<$ e $<=$

❖ $x > y$, vero se x assume un valore maggiore di y

❖ $x == y$, vero se x assume un valore uguale a y

❖ $x != y$, vero se x assume un valore diverso da y



Espressioni

❖ Operatori Logici:

➤ && (and), || (or), ! (not)

x	y	x&& y	x y	! x
0	0	0	0	Vero
0	Vero	0	Vero	Vero
Vero	0	0	Vero	0
Vero	Vero	Vero	Vero	0

Tavola di verità degli operatori logici, dove **0** corrisponde a **FALSO**



Espressioni

❖ Combinazione tra Operatori Relazionali e Logici

❖ Esempi:

- **$x==y \ \&\& \ a>b$** è vera se (x è uguale a y) e contemporaneamente (a è maggiore di b)
- **$i>n \ \&\& \ x!=y$** è vera se (i è maggiore di n) e contemporaneamente (x è diversa da y)
- **$b<c \ || \ t!=r$** è vera solo se (b è minore di c) **OPPURE** (t è diversa da r); se valgono entrambe è ancora vera

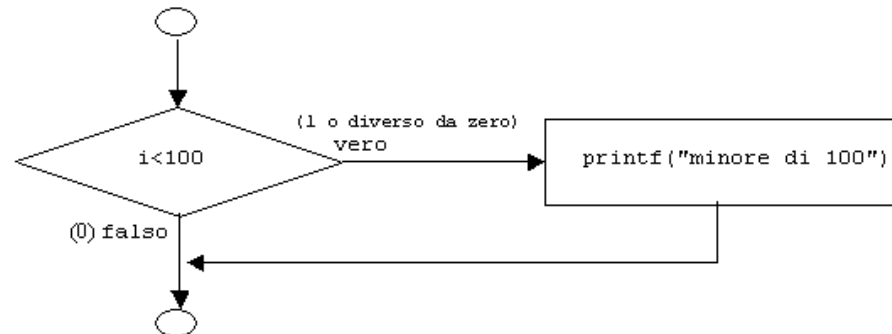
Istruzioni Condizionali: if

Per eseguire un'istruzione al verificarsi di una certa condizione, si utilizza l'istruzione `if`. Sintassi:

`if(espressione) istruzione`

- ❖ se ***espressione*** è vera si esegue ***istruzione***;
- ❖ L'istruzione può essere un blocco
- ❖ ***espressione*** restituisce ***vero*** o ***falso***;
- ❖ Una espressione che fornisce valore 0 è falsa
- ❖ Una espressione che fornisce un valore diverso da 0 è vera
- ❖ Esempio: Per visualizzare minore di 100 se `i` è minore di 100:

```
if(i<100) printf("minore di 100");
```





Istruzioni Condizionali: if

❖ Classico Errore:

```
if(i<100); printf("Valore minore di 100");
```

❖ Uso di Blocchi:

```
if(i<100) printf("Valore minore di 100");  
printf("il Valore di i e' %d ",i);
```

```
if(i<100) {  
    printf("Valore minore di 100");  
    printf("il Valore di i e' %d ",i);  
}
```



Istruzioni Condizionali: if-else

- ❖ La sintassi completa dell'istruzione `if` è:

```
if (espressione) istruzione1 else istruzione2
```

se *espressione* è vera viene eseguita *istruzione1*,

altrimenti (*else*) se è falsa viene eseguita *istruzione2*.

- ❖ Nulla vieta che una delle due istruzioni sia la stessa istruzione **if**

```
if (a>=10) printf("a maggiore o uguale a 10\n");
```

```
else if (a>=5) printf("a compreso tra 5 (incluso) e 10 (escluso) \n");
```



Istruzioni Condizionali: if-else

- ❖ La sintassi completa dell'istruzione `if` è:

```
if (espressione) istruzione1 else istruzione2
```

se *espressione* è vera viene eseguita *istruzione1*,

altrimenti (*else*) se è falsa viene eseguita *istruzione2*.

- ❖ Nulla vieta che una delle due istruzioni sia la stessa istruzione **if**

```
if (a>=10) printf("a maggiore o uguale a 10\n");
```

```
else if (a>=5) printf("a compreso tra 5 (incluso) e 10 (escluso) \n");
```

```
else if (a>=0) printf("a compreso tra 0 (incluso) e 5 (escluso) \n");
```

Istruzioni Condizionali: if annidati

❖ **Supponiamo di voler tradurre l'algoritmo:**

se $i < 100$, nel caso in cui $i > 0$ scrivi sul video la frase **"minore di 100 e maggiore di zero"**;

invece se $i \geq 100$ scrivi sul video la frase **"maggiore o uguale a 100"**;

❖ **Errore:**

```
if (i < 100)  
  if (i > 0)  
    printf("minore di 100 e maggiore di zero");  
  else printf("maggiore o uguale a 100");
```

❖ **Perché ?:** **il ramo else si riferisce all'if più interno**

Istruzioni Condizionali: if annidati



❖ Soluzione: uso dei blocchi

```
if (i<100) {
```

```
    if (i>0)
```

```
        printf("minore di 100 e maggiore di zero");
```

```
    };  
else printf("maggiore o uguale a 100");
```

Istruzioni Condizionali: if annidati

Classici Errori:

```
if(i<100)
    printf("numero minore ");
    printf("di 100"); ← Errore in Compilazione
else
    printf("numero maggiore ");
    printf("o uguale a 0"); ← Errore logico
```

Versione Corretta:

```
if(i<100) {
    printf("numero minore ");
    printf("di 100");
} else {
    printf("numero maggiore ");
    printf("o uguale a 0");
}
```

Esercizio: if-else

```
#include <stdio.h>

int i;

int main (void)
{
    printf("Dammi un intero: ");
    scanf("%d", &i);

    if (i<100)
        printf("\nminore di 100");
    else
        printf("\nmaggiore o uguale a 100");
}
```

Esercizio: if-else



```
#include<stdio.h>

double n;

int main(void) {

    printf("\nInserisci un numero diverso da zero ");
    scanf("%lf",&n);

    if (!n) printf("\nErrore\n");
    else printf("\n%f",1/n);
}
```


Esercizio: if-else

65	'A'	97	'a'
66	'B'	98	'b'
67	'C'	99	'c'
68	'D'	100	'd'
69	'E'	101	'e'
70	'F'	102	'f'
71	'G'	103	'g'

```
#include<stdio.h>

char n;

int main(void)
{
    printf("\nInserisci una lettera dell'alfabeto ");
    scanf("%c",&n);

    if (n>='a' && n<='z') {
        printf("\nCarattere Minuscolo");
        printf("\nIl Carattere Maiuscolo e' = %c ",'A'-'a'+n);
    } else if (n>='A' && n<='Z') {
        printf("\nCarattere Maiuscolo");
        printf("\nIl Carattere Minuscolo e' = %c ",'a'-'A'+n);
    } else printf("\nNon hai inserito una lettera dell'alfabeto");
}
```

Esercizio:

if-else

Per verificare se il carattere è una lettera scriviamo:

```
if(x>='A' && x<='Z' || x>='a' && x<='z')
    printf("\nSi tratta di una lettera");
else
    printf("\nNon si tratta di una lettera");
```



Esercizio: if-else

Data l'equazione $ax + b = 0$,
scrivere un programma in linguaggio C per
determinare il valore di x , se esiste, che risolve
l'equazione: $x = \frac{-b}{a}$

```
#include <stdio.h>
float a, b, x;

int main(void) {

    printf("\nRisoluzione equazioni di primo grado ");
    printf("\nEquazione nella forma: ax + b = 0 ");
    printf("\nImmetti coefficiente a: ");
    scanf("%f", &a);
    printf("\nImmetti coefficiente b: ");
    scanf("%f", &b);

    if(a) {
        x = - b / a ;
        printf("\nLa soluzione e' x = %f ", x) ;
    } else if( !b )
        printf("\nEquazione indeterminata (infinite soluzioni)");
    else printf("\nEquazione impossibile (non ammette soluzioni) ");
}
```

Istruzioni Condizionali: switch-case

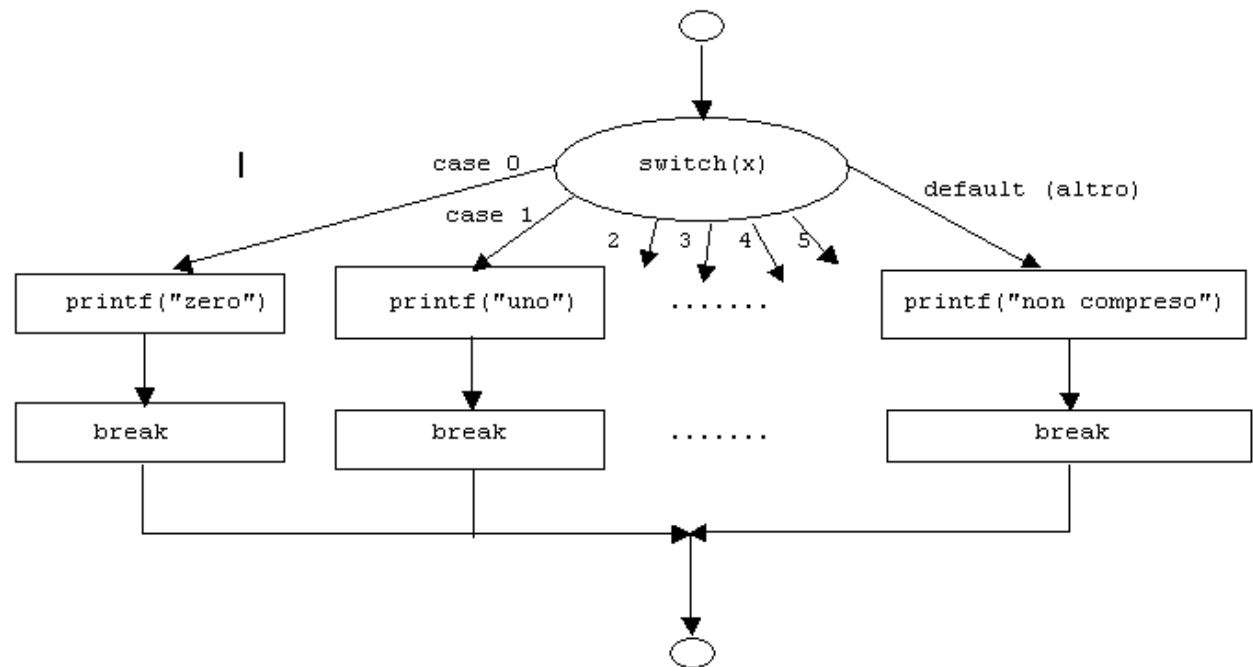
- ❖ Implementa **decisioni multiple** confrontando un'*espressione* (**int o char**) e un insieme di valori costanti.
- ❖ Ad ogni valore costante è associata una o più istruzioni
- ❖ La parola chiave **case** precede ogni valore costante
- ❖ Il numero di case non è limitato
- ❖ Se *espressione* è uguale ad uno dei valori costanti il controllo passa alla sequenza di istruzioni a partire dalla prima istruzione associata al valore costante
- ❖ Oltre alla parola chiave **case**, si può utilizzare (opzionale) anche una e una sola parola chiave **default**
- ❖ Se l'espressione non corrisponde a nessuna delle costanti, il controllo è trasferito alla prima istruzione successiva a default, se presente. Altrimenti si esce dal blocco switch

```
switch(espressione) {  
    case costante1:  
        istruzione  
        ...  
    case costante2:  
        istruzione  
        ...  
    case costante3:  
        istruzione  
        ...  
    default:  
        istruzione  
        ...  
}
```

Istruzioni Condizionali: switch-case

Utilizzo del comando
break all'interno
dello switch

```
switch(espressione) {  
  case costante1:  
    istruzione  
    ...  
    break;  
  case costante2:  
    istruzione  
    ...  
    break;  
  case costante3;  
    istruzione  
    ...  
    break;  
  ...  
  [default:  
    istruzione  
    ...    ]  
}
```



Istruzioni Condizionali: switch-case

```
#include <stdio.h>
int x;

int main(void) {
    printf("Digita una cifra: ");
    scanf("%d", &x);
    switch(x) {
        case 0:
            printf("zero\n");
            break;
        case 1:
            printf("uno\n");
            break;
        case 2:
            printf("due\n");
            break;
        case 3:
            printf("tre\n");
            break;
        default:
            printf("non compreso\n");
    }
}
```

Istruzioni Condizionali: switch-case

È anche possibile far corrispondere a un gruppo di istruzioni più costanti, ripetendo più volte la parola chiave `case` seguita dai due punti.

```
#include <stdio.h>

unsigned short x;

int main(void){
    printf("Digita una cifra tra 1 e 6: ");
    scanf("%d", &x);

    switch(x) {
        case 2:
        case 4:
        case 6:
            printf("\npari");
            break;
        case 1:
        case 3:
        case 5:
            printf("\ndispari");
            break;
        default:
            printf("\nnumero 0 o > 6");
    }
}
```

Istruzioni Condizionali: switch-case

Classico uso:
menu di scelta

```
printf("\n\t\t\tMenu\n");  
printf("\n1) Operazione 1");  
printf("\n2) Operazione 2");  
printf("\n3) Operazione 3");  
printf("\n4) Fine Programma ");  
printf("\nInserisci la tua scelta: ");  
scanf("%u",&scelta);
```

```
switch(scelta) {  
    case 1:  
        istruzione1;  
        break;  
    case 2:  
        istruzione2;  
        break;  
    case 3:  
        istruzione3;  
        break;  
}
```

Non è necessario

