

# **C: strutture di controllo decisionali**

Fondamenti di Programmazione

# Operatori relazionali

- `==` uguaglianza
- `!=` diversità
- `<` minore di
- `>` maggiore di
- `<=` minore o uguale a
- `>=` maggiore o uguale a

# Espressione logica

Espressione che genera un risultato `VERO` o `FALSO`

In C:

- `1` `VERO`
- `0` `FALSO`

Qualsiasi valore diverso da 0 viene valutato come `VERO`

# Operatori logici

Permettono di concatenare più espressioni logiche

- `!` not
- `&&` and
- `||` or

# Tavola di verità


x	y	x & y	x    y	! x
---	---	---	---	---
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

# Alcuni esempi utili

Le espressioni logiche restituiscono un valore numerico!

```
a = i<100;
```

a vale 1 se  $i < 100$ , vale 0 se  $i \geq 100$

**ATTENZIONE:** differenza tra operatore `=` e `==` 

# if

- Consente di eseguire un'istruzione al verificarsi di una condizione
- Oppure di eseguire **in alternativa** un'altra istruzione

```
if (condizione) {istruzioni} else {istruzioni}
```

else è opzionale

Se la condizione è seguita da **una sola istruzione** le parentesi graffe possono essere omesse ⚡

## if: esempio

```
if(x==0) printf("x vale 0");
```



# Esercizio

Scrivere un programma in C che permetta di valutare se un numero intero (letto da tastiera) è pari o dispari

# Soluzione

```
#include <stdio.h>

int main()
{
    int n;

    printf("Inserisci un numero intero: ");
    scanf("%d",&n);

    if(n%2 == 0)
        printf("\n%d e' pari\n",n);
    else
        printf("\n%d e' dispari\n",n);

    return 0;
}
```

# Esercizio

Scrivere un programma in C che permetta di valutare se un numero intero (letto da tastiera) è positivo o negativo

# Soluzione

```
#include <stdio.h>

int main()
{
    int n;

    printf("Inserisci un numero intero: ");
    scanf("%d",&n);
    if(n>=0)
        printf("\n%d e' positivo\n",n);
    else
        printf("\n%d e' negativo\n",n);

    return 0;
}
```

# Esercizio

Scrivere un programma in C che permetta di verificare se un anno (letto da tastiera) è bisestile.

(divisibile per 4 `&&` non divisibile per 100) `||` divisibile per 400

# Soluzione

```
#include <stdio.h>

int main()
{
    int anno;

    printf("Inserisci l'anno: ");
    scanf("%d",&anno);

    if ((anno%4 == 0 && anno%100 != 0) || anno%400 == 0)
        printf("\n%d e' bisestile\n", anno);
    else
        printf("\n%d non e' bisestile\n", anno);

    return 0;
}
```

# if annidati

```
//DUMMY EXAMPLE
#include <stdio.h>

int main()
{
    int test1=1,test2=2;
    if(test1==1)
        if(test2==2)
            printf("I test sono verificati\n");
        else
            printf("Test2 non e' verificato\n");
    else
        printf("Test1 non e' verificato\n");

    return 0;
}
```

Ambiguità: **else** è sempre associata all'ultima istruzione **if**  
senza else

# switch

- Il costrutto `if-else` talvolta risulta innaturale.
- Lo `switch` permette di selezionare un'istruzione (tra un insieme di istruzioni possibili) in base al valore assunto una variabile (o da un'espressione)
- Confronto tra risultato espressione ( `int` o `char` ) e un insieme di valori costanti ( `int` o `char` ).



# switch

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

# switch

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

# switch

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

```
#include <stdio.h>
int main()
{
    int giudizio;
    printf("Inserisci un giudizio tra 1-3\n");
    scanf("%d",&giudizio);
    switch (giudizio)
    {
        case 1:
            printf("Hai assegnato 1!\n");
            printf("Ci dispiace...\n");
            break;
        case 2:
            printf("Hai assegnato 2!\n");
            printf("Grazie!\n");
            break;
        case 3:
            printf("Hai assegnato 3!\n");
            printf("Grazie mille!.\n");
            break;
        default:
            printf("Giudizio non valido!\n");
    }
    return 0;
}
```

# Esercizio

Scrivere un programma in C che permetta di valutare se un numero intero è pari o dispari (si usi switch-case)

# Soluzione

```
#include <stdio.h>
int main()
{
    int n;
    printf("Inserisci un numero intero \n");
    scanf("%d",&n);

    switch (n % 2)
    {
        case 0:
            printf("Il numero %d e' pari\n",n);
            break;
        case 1:
            printf("Il numero %d e' dispari\n",n);
            break;
    }
    return 0;
}
```

# Esercizio

Creare una calcolatrice che presi due numeri da tastiera permetta di calcolare in base alla scelta effettuata dall'utente:

- somma
- sottrazione
- moltiplicazione
- divisione

# Soluzione

```
#include <stdio.h>
int main()
{
    int n1,n2;
    char oper;

    printf("Inserisci un primo numero \n");
    scanf("%d",&n1);

    printf("\nDefinisci l'operazione \n");
    printf("+ - * /\n");
    scanf(" %c",&oper);

    printf("\nInserisci un secondo numero \n");
    scanf("%d",&n2);
```



```
switch (oper)
{
    case '+':
        printf("%d %c %d: %d\n",n1,oper,n2,n1+n2);
        break;
    case '-':
        printf("%d %c %d: %d\n",n1,oper,n2,n1-n2);
        break;
    case '*':
        printf("%d %c %d: %d\n",n1,oper,n2,n1*n2);
        break;
    case '/':
        printf("%d %c %d: %d\n",n1,oper,n2,n1/n2);
        break;
    default:
        printf("Scelta non valida!\n");
}
return 0;
}
```

# Esercizio

Creare un programma in C che presi in ingresso il peso (in Kg) e l'altezza (in m) di una persona, calcoli il BMI.

$$BMI = peso / (altezza^2)$$

Il programma dovrà successivamente assegnare una delle seguenti tre classi:

- sottopeso se  $BMI \leq 20$
- normopeso se  $20 < BMI \leq 30$
- sovrappeso se  $BMI > 30$

# Soluzione

```
#include <stdio.h>
int main()
{
    float altezza,peso,bmi;
    printf("Inserisci altezza in m:\n");
    scanf("%f",&altezza);
    printf("Inserisci peso in Kg:\n");
    scanf("%f",&peso);
    bmi=peso/(altezza*altezza);
    if(bmi<=20) printf("BMI: %.2f sottopeso\n",bmi);
    else if (bmi>20 && bmi<=30)
        printf("BMI: %.2f Normopeso\n",bmi);
    else printf("BMI: %.2f Sovrappeso\n",bmi);
    return 0;
}
```

# Esercizio

Creare un programma in C che permetta di calcolare a scelta l'area di un cerchio, di un rettangolo o di un triangolo.

# Soluzione

```
#include <stdio.h>
int main()
{
    int sel,n,m;
    const float Pi=3.14;

    printf("Calcolo area\n");
    printf("1: cerchio; ");
    printf("2: rettangolo; ");
    printf("3: triangolo.\n");
    scanf("%d",&sel);
```

```

switch (sel)
{
    case 1:
        printf("Raggio: \n");
        scanf("%d",&n);
        printf("Area cerchio: %.2f\n",Pi*n*n);
        break;
    case 2:
        printf("Lati: \n");
        scanf("%d%d",&n,&m);
        printf("Area rettangolo: %d\n",n*m);
        break;
    case 3:
        printf("Base e altezza: \n");
        scanf("%d%d",&n,&m);
        printf("Area triangolo: %.2f\n",n*m*.5);
        break;
    default:
        printf("Scelta non valida!\n");
}
return 0;
}

```