

Unità F2

Selezione

© 2007 SEI-Società Editrice Internazionale, Apogeo

Obiettivi

- Conoscere il significato delle strutture di controllo decisionali
- Conoscere gli operatori matematici, di confronto e logici
- Conoscere ed essere in grado di utilizzare i dati di tipo carattere
- Essere in grado di realizzare programmi con selezioni, blocchi d'istruzioni, selezioni di selezioni
- Essere in grado di scambiare i valori di due variabili

© 2007 SEI-Società Editrice Internazionale, Apogeo

Problema

- **Problema:** verificare se i tre valori passati in ingresso sono una terna pitagorica.
- **Nota:** il primo valore immesso deve essere il maggiore dei tre.
- **Input:** tre valori numerici interi, il primo deve essere il maggiore dei tre.
- **Output:** in caso di verifica positiva, viene segnalato che si tratta di una terna pitagorica.

© 2007 SEI-Società Editrice Internazionale, Apogeo

Algoritmo

```

graph TD
    Start([Inizio]) --> Read1[/leggi valore1/]
    Read1 --> Read2[/leggi valore2/]
    Read2 --> Read3[/leggi valore3/]
    Read3 --> Calc["differenza = valore1^2 - valore2^2 - valore3^2"]
    Calc --> Dec{differenza = 0}
    Dec -- VERO --> Print[/scrivi "è una terna pitagorica"/]
    Dec -- FALSO --> End([Fine])
    Print --> End
  
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Commento all'algoritmo

- Il flusso di esecuzione non è più lineare.
- Nel blocco decisionale un'istruzione è eseguita solo al verificarsi di una certa condizione
- Nella programmazione strutturata i costrutti di controllo devono avere un solo punto di ingresso e un solo punto di uscita: questo vincolo è rispettato dalla struttura di controllo decisionale

© 2007 SEI-Società Editrice Internazionale, Apogeo

Selezione

```

/* Terna pitagorica */
Algoritmo TernaPitagorica
valore1, valore2, differenza di tipo Intero;
Se (<condizione>) Allora
  <istruzione>
FineSe
Pseudolingaggio: selezione

/* Terna pitagorica */
Algoritmo TernaPitagorica
valore1, valore2, differenza di tipo Intero;
Scrivi "verifica terna pitagorica"
Scrivi ("primo valore (il maggiore): ")
Leggi (valore1)
Scrivi ("secondo valore: ")
Leggi (valore2)
Scrivi ("terzo valore: ")
Leggi (valore3)
differenza ← valore1*valore1 - (valore2*valore2+valore3*valore3)
Se (differenza=0) Allora
  scrivi ("è una terna pitagorica")
FineSe
FineAlgoritmo
  
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Linguaggio C - if

- Ogni linguaggio di programmazione presenta almeno una **struttura di controllo decisionale**.
- La sintassi del linguaggio C è la seguente.
if (<espressione> <istruzione>;
- Si tratta di una **selezione**, all'interno della quale se e solo se **espressione** è vera viene eseguita l'istruzione.

Programmazione: IF
if (<espressione>
<istruzione>;

© 2007 SEI-Società Editrice Internazionale, Apogeo

Programma

```
/* Terna Pitagorica */
#include <stdio.h>

main()
{
    int valore1, valore2, valore3, differenza;

    printf("Verifica terna pitagorica\n\n");
    printf("Primo valore (il maggiore): ");
    scanf("%d", &valore1);
    printf("Secondo valore: ");
    scanf("%d", &valore2);
    printf("Terzo valore: ");
    scanf("%d", &valore3);

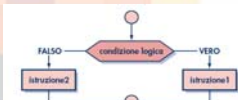
    differenza = valore1*valore1 - (valore2*valore2+valore3*valore3);

    if(differenza==0)
        printf("e' una terna pitagorica\n");
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

if - else

Pseudolinguaggio: Se ... Altrimenti ... FineSe
Se (<condizione>) **Allora**
 <istruzione1>
Altrimenti!
 <istruzione2>
FineSe



Programmazione: If ... Else in C
if (<espressione>
<istruzione1>;
else
<istruzione2>;

© 2007 SEI-Società Editrice Internazionale, Apogeo

Programma

```
Programma
/* Terna Pitagorica */
#include <stdio.h>

main()
{
    int valore1, valore2, valore3, differenza;

    printf("Verifica terna pitagorica\n\n");
    printf("Primo valore (il maggiore): ");
    scanf("%d", &valore1);
    printf("Secondo valore: ");
    scanf("%d", &valore2);
    printf("Terzo valore: ");
    scanf("%d", &valore3);

    differenza = valore1*valore1 - (valore2*valore2+valore3*valore3);

    if(differenza==0)
        printf("e' una terna pitagorica\n");
    else
        printf("non e' una terna pitagorica\n");
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Un esempio

Problema: verificare se un numero intero è pari o dispari.
Input: un numero intero.
Output: scrive se il numero ricevuto in ingresso è pari o dispari.
Programma risolutivo

```
/* Pari o dispari */
#include <stdio.h>
main()
{
    int numero;
    printf("Verifica se un numero è pari o dispari\n\n");
    printf("Immetti un numero intero: ");
    scanf("%d", &numero);
    if(numero%2==0)
        printf("è pari\n");
    else
        printf("è dispari\n");
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Operatori aritmetici

Operatore	Descrizione	Esempio	Descrizione esempio
-	Negazione unaria	-x	Il valore di x cambiato di segno
*	Prodotto	x*y	x moltiplicato y
/	Divisione	x/y	x diviso y
%	Resto della divisione intera	x%y	Resto della divisione intera di x per y
+	Addizione	x+y	x sommato a y
-	Sottrazione	x-y	x meno y

Priorità ↑

© 2007 SEI-Società Editrice Internazionale, Apogeo

Operatori di confronto

Tabella 2 Gli operatori di confronto

Operatori di confronto	Descrizione	Esempio	Descrizione esempio
>	maggiore	x>y	x maggiore di y
<	minore	x<y	x minore di y
>=	maggiore uguale	x>=y	x maggiore uguale di y
<=	minore uguale	x<=y	x minore uguale di y
==	uguale	x==y	x uguale a y
!=	diverso	x!=y	x diverso da y

© 2007 SEI-Società Editrice Internazionale, Apogeo

Operatori logici

Tabella 3 Gli operatori logici

Operatori logici	Descrizione	Esempio	Descrizione esempio
!	NOT logico	!x	NOT x
&&	AND logico	x&& y	x AND y
	OR logico	x y	x OR y

nota bene!

Vero e falso sono due valori booleani che in C/C++ sono rappresentati dai valori interi 1 e 0 (zero).

© 2007 SEI-Società Editrice Internazionale, Apogeo

Scambio di valori fra due variabili

Per scambiare i valori di due variabili è necessario utilizzare una variabile di appoggio, in cui salvare momentaneamente il valore di una di esse, prima di assegnare il valore dell'altra.

Stato iniziale delle variabili	valore2	valore1	scambio
	25	24	
scambio = valore1;	25	24	24
valore1 = valore2;	25	25	24
valore2 = scambio;	24	25	24

© 2007 SEI-Società Editrice Internazionale, Apogeo

Blocchi di istruzioni

- dove **bloccoIstruzioni** è costituito da un insieme d'istruzioni qualsiasi raccolte tra parentesi graffe:

Programmazione: If ... [Else]

```
if(<espressione>
  <bloccoIstruzioni>
[else <bloccoIstruzioni>]
```

```
if(<espressione>)
{
  <istruzione1>;
  <istruzione2>;
  ...
  <istruzioneN>;
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

```
/* Terna Pitagorica */
#include <stdio.h>
main()
{
  int valore1, valore2, valore3,
  differenza, scambio;

  printf("Verifica terna pitagorica\n\n");
  printf("Primo valore: ");
  scanf("%d", &valore1);
  printf("Secondo valore: ");
  scanf("%d", &valore2);
  printf("Terzo valore: ");
  scanf("%d", &valore3);

  /* scambio valore1 con valore2 */
  if(valore1<valore2)
  {
    scambio = valore1;
    valore1 = valore2;
    valore2 = scambio;
  }
  /* scambio valore1 con valore3 */
  if(valore1<valore3)
  {
    scambio = valore1;
    valore1 = valore3;
    valore3 = scambio;
  }

  differenza = valore1*valore1 - (valore2*valore2+valore3*valore3);
  if(differenza==0)
    printf("e' una terna pitagorica\n");
  else
    printf("non e' una terna pitagorica\n");
}
```

Selezione di selezioni

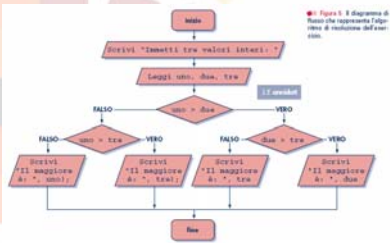
Problema: Determinare il maggiore di tre numeri.
Input: tre valori numerici interi.
Output: scrivere il maggiore.

```
/* Il maggiore di tre valori */
#include <stdio.h>
main() {
  int uno, due, tre;
  printf("Inserisci tre valori interi: ");
  scanf("%d %d %d", &uno, &due, &tre);

  if(uno>due)
    if(uno>tre)
      printf("Il maggiore e': %d\n", uno);
    else
      printf("Il maggiore e': %d\n", tre);
  else
    if(due>tre)
      printf("Il maggiore e': %d\n", due);
    else
      printf("Il maggiore e': %d\n", tre);
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Diagramma



© 2007 SEI-Società Editrice Internazionale, Apogeo

Il tipo di dato carattere

- Oltre ai tipi **int**, **float** e **double** esiste anche il tipo **carattere**:
- In linguaggio C la dichiarazione di variabili di tipo carattere è la seguente
`char lettera, cifra, simbolo;`
- char** è una **parola chiave** del linguaggio, utilizzata per definire le variabili di tipo **carattere** (**char** sta infatti per *character*), che possono contenere valori alfanumerici come lettere dell'alfabeto, cifre decimali, simboli di punteggiatura ecc.

© 2007 SEI-Società Editrice Internazionale, Apogeo

I caratteri in memoria

- Alla variabile viene riservato uno spazio di memoria atto ad accogliere un carattere alfanumerico.
- Un byte consente di rappresentare 256 caratteri, esattamente quanti ne comprende il **codice ASCII** (*American Standard Code for Information Interchange*) che include tutti i caratteri delle lingue latine oltre a molti caratteri speciali.
- Due byte consentono di rappresentare l'**Unicode**, che comprende i caratteri di tutte le lingue comprese quelle orientali come il Cinese.
- I codici fanno corrispondere a ogni carattere la loro rappresentazione numerica, quindi i caratteri risultano ordinati ed è possibile confrontarli.
- L'assegnamento a una variabile prevede che il valore sia compreso tra apici singoli:


```
a = 'F';
b = '4';
c = '?';
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Gli operatori sui caratteri

Operatore di confronto	Descrizione	Esempio	Descrizione esempio
>	maggiore	a > b	a maggiore di b
<	minore	a < b	a minore di b
>=	maggiore uguale	a >= b	a maggiore uguale di b
<=	minore uguale	a <= b	a minore uguale di b
==	uguale	a == b	a uguale a b
!=	diverso	a != b	a diverso da b

© 2007 SEI-Società Editrice Internazionale, Apogeo

switch-case

- Per le scelte multiple un'alternativa agli **if-else** in cascata è la struttura di controllo **switch-case**
- switch** funziona come un interruttore a posizioni multiple, se il valore di **espressione** è uguale a (**case**) **costante**, sono eseguite le istruzioni che seguono **costante**.
- Se nessuna **costante** è uguale a **espressione** e se è presente **default** sono eseguite le istruzioni successive a **default**.
- espressione** deve restituire un valore di tipo **int** o **char**.

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

© 2007 SEI-Società Editrice Internazionale, Apogeo

La struttura di switch - case

- La struttura è composta dalla parola chiave **switch** che ha l'effetto di valutare **espressione**.
- Se **espressione** è uguale a una certa **costante** sono eseguite tutte le istruzioni del blocco a partire da quella posizione.
- Se **espressione** non è uguale ad alcuna costante, sono eseguite le sole istruzioni della parte opzionale **default**.
- Normalmente quel che si desidera è che siano eseguite le sole istruzioni relative a un **case**, non tutte quelle che seguono. A tale scopo è disponibile l'istruzione **break** che provoca l'uscita dallo **switch**.

© 2007 SEI-Società Editrice Internazionale, Apogeo

Problema

- **Problema:** visualizzare le note musicali immesse in formato americano (A = la, B= si, C = do, D = re, E = mi, F = fa, G = sol).
- **Input:** una lettera dell'alfabeto.
- **Output:** visualizza la nota, se non compresa visualizza "La nota non esiste".

© 2007 SEI-Società Editrice Internazionale, Apogeo

Programma risolutivo

```
/* Visualizza le note immesse in formato americano */
#include <stdio.h>
main()
{
    char nota;
    printf("Immetti una nota in formato americano: ");
    scanf("%c", &nota);
    switch(nota)
    {
        case 'A':
            printf("la\n"); break;
        case 'B':
            printf("si\n"); break;
        case 'C':
            printf("do\n"); break;
        case 'D':
            printf("re\n"); break;
        case 'E':
            printf("mi\n"); break;
        case 'F':
            printf("fa\n"); break;
        case 'G':
            printf("sol\n"); break;
        default:
            printf("La nota non esiste\n");
    }
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo

Problema

- **Problema:** a seconda della posizione dell'interruttore 1, 2 o 3, si desiderano accendere 1, 2 o 3 lampadine. Tutte le altre posizioni non provocano l'accensione di alcuna lampadina.
- **Input:** un valore numerico intero.
- **Output:** ogni volta che accende una lampadina lo scrive oppure scrive "Buio".

© 2007 SEI-Società Editrice Internazionale, Apogeo

Programma risolutivo

```
/* Accendi il lampadario */
#include <stdio.h>
main()
{
    int interruttore;
    printf("Vuol accendere 1, 2 o 3 lampadine: ");
    scanf("%d", &interruttore);

    switch(interruttore)
    {
        case 3:
            printf("Accendo la terza\n");
        case 2:
            printf("Accendo la seconda\n");
        case 1:
            printf("Accendo la prima\n");
            break;
        default:
            printf("Buio\n");
    }
}
```

© 2007 SEI-Società Editrice Internazionale, Apogeo