



**UNIVERSITÀ DEGLI STUDI
DEL SANNIO** Benevento
DING
DIPARTIMENTO DI INGEGNERIA

CORSO DI "PROGRAMMAZIONE I"

Prof. Franco FRATTOLILLO
Dipartimento di Ingegneria
Università degli Studi del Sannio


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 1



Tipi strutturati

- L'informazione può essere decomposta in tipi più semplici
- Più informazioni possono essere aggregate fra loro in base ad una relazione per costituire un'informazione più complessa
- Es:
 - data (giorno, mese, anno)
 - numero complesso (parte reale, coefficiente immaginario)
 - generalità anagrafiche (cognome, nome, data nascita, indirizzo)
- Un tipo strutturato è caratterizzato da:
 - tipo componenti
 - costruttore
 - funzione d'accesso agli elementi


Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 2



Tipi strutturati

- tipi componenti (... tipi base ...)
- modalità di aggregazione (operazioni di)
- operazioni caratteristiche
 - funzioni di accesso (sono operazioni esterne al tipo strutturato che trasformano un oggetto del tipo in un suo componente)
 - altre operazioni ...
- nei linguaggi di programmazione
 - tipi strutturati primitivi
 - costruttori di tipo strutturato
 - dichiarazioni di informazioni di tipo strutturato
 - costruttori di valori

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 3



Array

- È praticamente presente in tutti i linguaggi come tipo strutturato generico e primitivo
- Metodo di aggregazione: Cartesiano

$$\text{tipo } \mathbf{array} = T_1 \times T_2 \times \dots \times T_n$$

$$T_1 = T_2 = \dots = T_n = T$$

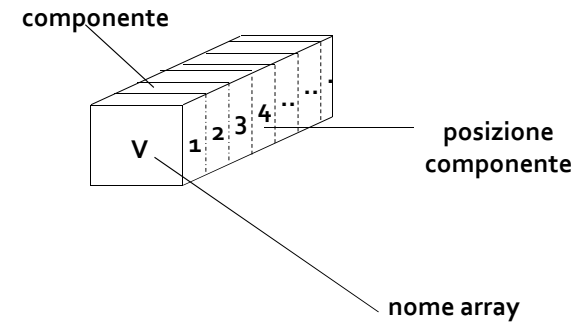
- ennuple ordinate di valori dello stesso tipo ...
- quando usato, bisogna:
 - definire T (il tipo dei componenti)
 - definire n (dimensione o cardinalità)

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 4

Funzione di Accesso

- Il cartesiano introduce una relazione d'ordine fra i componenti
 - ogni componente è individuabile attraverso la definizione della POSIZIONE che occupa nell'ordinamento
 - modalità di accesso Casuale o Diretta
 - le costanti, le informazioni, le espressioni usate per definire la posizione dell'elemento di interesse in operazioni di accesso ad un array vengono detti INDICI

... un modello interpretativo ...



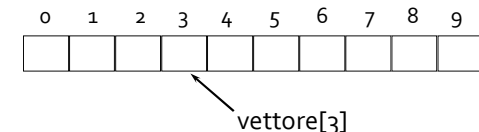
Array in C

- In C la dichiarazione di una variabile di tipo array è composta di quattro parti:
 - uno specificatore del tipo base
 - un identificatore della variabile
 - una coppia di parentesi quadre `[]`, che racchiude la dimensione (un valore costante) dell'array
- Dichiarazione: `tipo_base nome_array[dimensione];`
Esempio: `double vettore[10];`
- Possibile inizializzare un array e dichiarare costanti ...
 - Inizializzazione: `int vettore[5] = {4, 6, 3, 7, 9};`
 - `int vettcost[5] = {1, 3, 5, 7, 9};`

Array in C: funzione di accesso

- Se un array ha dimensione N i suoi elementi sono indicizzati da 0 a $N-1$
- Individuazione degli elementi dell'array:
 - `nomeArray[posizione]`
 - Esempi:


```
double vettore[10];
vettore[3] = 5;
x = vettore[4] * 2;
```



Indice

- Appartiene ad un tipo ordinato e indica la posizione di un elemento nello array
- Può essere espresso tramite:
 - una costante (es. $A[4]$, quarto elemento dello array)
 - una variabile (es. $A[i]$, i-esimo elemento dello array)
 - un'espressione valutata intera (es. $A[c+d*e]$ (elemento dello array nella posizione corrispondente al valore di $c+d*e$))

Assegnamento degli elementi di un array

- Ad un elemento di un array è possibile assegnare un valore tramite operazioni di:
 - input
 - Es.: `scanf("%d", &A[3]);`
 - legge un valore che viene assegnato al terzo elemento dell'array A
 - Calcolo e assegnamento
 - Es.:


```
A[4] = C;
A[1] = 5;
A[3] = f * 9 + d;
```

Uso degli elementi di un array

- Un elemento di un array può essere utilizzato in operazioni di:
 - stampa
 - Es.: `printf("%d", A[3]);`
 - stampa il valore dell'elemento nel terzo elemento dello array A
 - calcolo e assegnazione
 - Es.:


```
C = A[4];
B = f * 9 + d - A[i];
```
 - per la valutazione di una condizione
 - Es.: `if (A[i] > 7)`

...

 $B = A[2] + A[4]; \Rightarrow B = 5 + 21 = 26$
`printf("%d --- %d\n", A[1], A[5]);`

22 --- 6

0	18
1	22
2	5
3	8
4	21
5	6

A

 $A[1] = 22; \Rightarrow$

0	18
1	22
2	5
3	8
4	21
5	6

A

Esempio: calcolo del punteggio medio

```
#include <stdio.h>
void main( ) {
    double punteggio[5], totale, media;
    int i;
    for(i=0; i<5; i++) {
        printf("Punteggio del giudice %d: ", i);
        scanf ("%f", &punteggio[i]);
    }
    totale=0;
    for(i=0; i<5; i++)
        totale=totale+punteggio[i];
    media=totale/5;
    printf("Il punteggio medio e' %f\n", media);
}
```

Fase di lettura
dei dati

Fase di elaborazione
dei risultati

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT

13

Esempio di esecuzione

```
for (i = 0; i < Ngiudici; i++) {
    printf("Punteggio del giudice %d: ", i);
    scanf ("%f", &punteggio[i]);
}
```

Ngiudici = 5

i	
0	9
1	8
2	7
3	8
4	10

punteggio

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT

14

Esempio di esecuzione

```
totale = 0;
for (i = 0; i < Ngiudici; i++) totale = totale+punteggio[i];
media = totale / Ngiudici;
```

i	punteggio	totale
0	9	9
1	8	17
2	7	24
3	8	32
4	10	42

media = 8.4

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT

15

Cardinalità e riempimento

- In C è necessario dichiarare la dimensione dell'array (affinché il compilatore possa allocare la memoria necessaria)
- Per generalizzare i programmi ad array di dimensione variabile, bisogna definire una dimensione massima *max* e, di volta in volta, utilizzare un numero di elementi *n* compreso tra 1 e *max*
- I valori *max* e *n* vengono detti rispettivamente cardinalità e riempimento dell'array

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio

Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT

16

Esempio rivisto

```
#include <stdio.h>
void main( ) {
    double punteggio[10], totale, media;
    int i, Ngiudici;
    do {
        printf("Numero di giudici: "); scanf("%d", &Ngiudici);
    } while(Ngiudici<1 || Ngiudici>10);
    for(i=0; i<Ngiudici; i++) {
        printf("Punteggio del giudice %d: ", i); scanf("%f", &punteggio[i]);
    }
    totale=0;
    for(i=0; i<Ngiudici; i++) totale=totale+punteggio[i];
    media=totale/Ngiudici;
    printf("Il punteggio medio e' %f\n", media);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 17

Somma dei numeri di indice dispari

- Definizione del problema: inserire i valori degli elementi in un array monodimensionale di interi, composto da 50 elementi; visualizzare il contenuto dell'array così ottenuto; calcolare la somma degli elementi di indice dispari
- Definizione dei dati del problema:
 - I: il valore di ciascun elemento dell'array di numeri interi
 - Precondizione: il numero degli elementi da inserire non può essere maggiore della cardinalità dell'array
 - U: l'array "riempito" con i valori degli elementi, la somma degli elementi di posto dispari
 - Postcondizione: nessuna

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 18

Somma dei numeri di indice dispari

- Descrizione del metodo di elaborazione:
 - Con un ciclo ripetitivo, si effettua la lettura del valore di ciascun elemento dell'array
 - Si visualizza, sempre usando un ciclo ripetitivo, il contenuto dell'array, stampando, ad ogni passo del ciclo, il valore di ciascun elemento immesso
 - Con un ciclo ripetitivo che parte dalla posizione 1 e con step di incremento pari a due si calcola la sommatoria dei valori nelle posizioni di indice dispari
 - Si stampa il risultato ottenuto

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 19

Il codice

```
#include<stdio.h>
main( ) {
    int Vett[50], i, Sommadispari;
    for(i = 0; i < 50; i++) {
        printf("Valore di Vett(%d): ", i); scanf("%d", &Vett[i]);
    }
    for(i = 0; i < 50; i++) printf("Vett(%d)=%d\n", i, Vett[i]);
    Sommadispari=0;
    for(i = 1; i < 50; i+=2) Sommadispari = Sommadispari+Vett[i];
    printf("Sommatoria valori elementi ");
    printf("in posti di indice dispari=%d\n", Sommadispari);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 20

Ricerca del minimo

- Trovare il valore minimo tra gli elementi di un array di interi, con cardinalità 50, ed indicarne la posizione
- Definizione dei dati del problema:
 - I: il riempimento; il valore di ciascun elemento
 - Pi: il numero degli elementi da inserire non può essere maggiore della cardinalità dell'array
 - U: la posizione dell'elemento con valore minimo
 - Pu: nessuna

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 21

Ricerca del minimo

- Descrizione del metodo di elaborazione:
 - È indicato il numero di quanti elementi si vogliono inserire: tale numero deve essere minore della cardinalità dell'array
 - Si legge il valore di ciascun elemento da immettere
 - Si pone pari al minimo il primo elemento dell'array
 - Si verifica se ciascun altro elemento dell'array è inferiore al minimo fissato; in tal caso, si aggiorna il valore del minimo e della posizione in cui esso si trova nell'array
 - Si stampa il valore minimo e la posizione in cui esso si trova

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 22

Ricerca del minimo

```

#include<stdio.h>
main( ) {
    const int CARDINALITA = 50;
    int VET[CARDINALITA], I, riemp, MIN, POS;
    do {
        printf ("Quanti elementi vuoi inserire? ");   scanf ("%d", &riemp);
    } while((riemp <= 0) || (riemp > CARDINALITA));
    for(I = 0; I < riemp; I++) {
        printf("Inserisci elemento di posto %d: ", I);   scanf("%d", &VET[I]);
    }
    MIN=VET[0];   POS=0;
    for(I = 1; I < riemp; I++)
        if(VET[I] < MIN) {
            MIN=VET[I];   POS=I;
        }
    printf("Il valore minimo = %d si trova nella posizione numero %d\n", MIN, POS);
}
  
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 23

Ricerca di un elemento

- Verificare se in un array monodimensionale di interi, composto da 50 elementi, esiste un determinato valore immesso in input. Se questo valore esiste, indicare la posizione del primo elemento che lo contiene e terminare la ricerca
- Definizione dei dati del problema:
 - I: il valore di ciascun elemento; il valore da cercare; (eventuale) riempimento
 - Pi: se usato un riempimento questo non può essere maggiore della cardinalità dell'array
 - U: la posizione del primo elemento con valore pari a quello cercato
 - Pu: nessuna

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 24

Ricerca di un elemento

- **Descrizione del metodo di elaborazione:**
 - È indicato il numero di quanti elementi si vogliono inserire: tale numero deve essere minore della cardinalità dell'array
 - Con un ciclo ripetitivo, si legge il valore di ciascun elemento da immettere
 - Si immette il valore da cercare nell'array
 - Con un ciclo iterativo, si verifica se tale valore esiste nell'array: se esiste, viene visualizzata la posizione del primo elemento che lo contiene
 - Se non esiste, viene visualizzata la scritta "il valore indicato non esiste nello array"

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 25

Ricerca di un elemento

```
#include<stdio.h>
main( ) {
    const int cardinalita = 50;
    int VET[cardinalita], I, riemp, VAL, POS, trovato;
    do {
        printf("Quanti elementi vuoi inserire? ");
        scanf("%d", &riemp);
    } while((riemp <= 0) || (riemp > cardinalita));
    for(I = 0; I < riemp; I++) {
        printf("Inserisci l'elemento di posto %d: ", I);
        scanf("%d", &VET[I]);
    }
    printf("Immetti il valore da ricercare: ");
    scanf("%d", &VAL);
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 26

...

```
.....
trovato=0; POS=0; I=0;
do {
    if(VET[I]==VAL) {
        trovato=1; POS=I;
    }
    I++;
} while((I < riemp) && (trovato != 1));
if(trovato==1)
    printf("Valore immesso %d in posizione %d\n", VAL, POS);
else
    printf("Il valore indicato %d non esiste nello array\n", VAL);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 27

... un'altra soluzione

```
.....
POS=0;
I=0;
while((I < riemp) && (VET[I] != VAL)) I++;
if(I < riemp) {
    POS=I;
    printf("Valore immesso %d in posizione %d\n", VAL, POS);
}
else
    printf("Il valore indicato %d non esiste nello array\n", VAL);
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 28

... ancora un'altra soluzione

```

.....
trovato=0;
I=0;
while((I<riemp) && (trovato==0)) {
    trovato=(VET[I]==VAL);
    I++;
}
if(trovato==1)
    printf("Il valore immesso %d in posizione %d\n", VAL, (I-1));
else
    printf("Il valore indicato %d non esiste nello array\n", VAL);
}

```

Array multidimensionali

- Gli array multidimensionali vengono dichiarati specificando il numero di elementi per ciascuna dimensione
- Un array bidimensionale con 6 elementi per ciascuna dimensione viene dichiarato come:
 - `int alfa[2][6];`
- Per referenziare un singolo elemento è necessario utilizzare due coppie di parentesi quadre:
 - `alfa[1][2] = 1;`
- In pratica, un array multidimensionale è una collezione di oggetti, ciascuno dei quali è un array

Esempio

```

/* Tabelline dall'1 al 10 */
main( ) {
    int tabelline[10][10];
    int i, j;
    for(i = 0 ; i < 10 ; i++)
        for(j = 0 ; j < 10 ; j++)
            tabelline[i][j] = (i + 1) * (j + 1);
}

```

Array: inizializzazione

- Gli array non sono inizializzati a zero: all'atto della loro definizione contengono valori indefiniti
- È possibile inizializzare esplicitamente un array racchiudendo i valori tra parentesi graffe e separandoli da virgole; in questo caso non è necessario specificare la dimensione dell'array
- Per gli array multidimensionali è invece necessario specificare tutte le dimensioni meno la prima

Esempio

```
int array_1[ ] = {0,1,2,3,4,5,6,7,8,9};
int array_2[ ][4] = {
    {1,2,3,4},
    {5,6,7,8},
    {122,144,678,908}
};
/* che è equivalente a */
int array_2[3][4] = {
    {1,2,3,4},
    {5,6,7,8},
    {122,144,678,908}
};
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 33

Operazioni sugli array

- Per copiare elementi da un array all'altro bisogna copiare singolarmente ogni elemento

```
int a[15], b[15], i;
for(i = 0 ; i < 15 ; i++) a[i] = b[i];
```

- È illegale invece assegnare

```
a = b;
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 34

Esempio: somma di matrici

```
main( ) {
    int matrix1[ ][3] = { {1,2,3}, {4,5,6}, {7,8,9} };
    int matrix2[ ][3] = { {3,4,2}, {2,1,4}, {4,6,9} };
    int i, j, result[3][3];
    for(i=0 ; i < 3 ; i++)
        for(j=0 ; j < 3 ; j++)
            result[i][j] = matrix1[i][j] + matrix2[i][j];
}
```

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 35

Array di caratteri

- Una stringa è un array monodimensionale di caratteri ASCII terminati da un carattere 'NULL' pari a '\0'
- Ad esempio: "Questa è una stringa" è un array di 21 caratteri
- L'array è quindi il seguente:

elemento zero	'Q'
primo elemento	'u'
secondo elemento	'e'
.....	
ventesimo elemento	'a'
ventunesimo elemento	'\0'

Franco FRATTOLILLO - Dipartimento di Ingegneria - Università degli Studi del Sannio Corso di "Programmazione I" - Corso di Laurea in Ingegneria Informatica / ExAT 36



Esempio

/* Stampa i caratteri e la codifica ASCII */

```
char stringa[ ] = {'a', 'A', 'H', 'k', '\0'};
main( ) {
    int i = 0;
    while( stringa[i] != '\0' ) {
        printf("%c = %d\n", stringa[i], stringa[i] );
        ++i;
    }
}
```



Esempio

/* Legge una stringa in input di max 100 caratteri */

```
main( ) {
    char linea[101];
    int c, i;
    for(i=0; (i<100) && ((c=getchar( )) != '\n'); i++) linea[i] = c;
    linea[i] = '\0';
    i = 0;
    while(linea[i] != '\0') printf("%c", linea[i++]);
}
```



Esempi

```
int i;
char c;
```

/* Valuta se c è una cifra */

```
i = (c >= '0' && c <= '9') ? 1 : 0;
```

/* Valuta se c è uno spazio */

```
i = (c == ' ') ? 1 : 0;
```



Esempio

/* trasforma i caratteri maiuscoli in minuscoli */

```
char str[100];
int i;
... /* inizializzazione di str con una stringa */
for(i=0; str[i] != '\0'; i++)
    if ((str[i] >= 'A') && (str[i] <= 'Z'))
        str[i] = str[i] + 'a' - 'A';
```



Costanti stringa

- Una costante stringa è una sequenza di uno o più caratteri racchiusi da doppi apici ("")
 - il compilatore aggiunge un carattere NULL ('\0') alla fine di ogni stringa
- costanti stringa separate da spazi/ritorni a capo sono concatenate
 - il carattere " (doppio apice) in una stringa va preceduto da \

```
char a[]="Ciao Mondo";
```

```
char a[]="Ciao" " Mondo";
```

```
char a[]="Ciao"
```

```
        "Mondo";
```

```
char a[]="Titolo libro: \"Divina Commedia\"";
```