

Capitolo 6 - Array

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Array

- Array
 - Gruppo di locazioni di memoria consecutive
 - Stesso nome e tipo
- Per riferirsi a un elemento, specificare
 - Nome dell'array
 - Posizione
- Formato:
 - arrayname[position number]*
 - Primo elemento a posizione 0
 - array di n elementi di nome c:
 - $c[0], c[1] \dots c[n - 1]$

Nome dell'array
(Tutti gli elementi di questo array hanno lo stesso nome, c)

c[0]	-45
c[1]	6
c[2]	0
c[3]	72
c[4]	1543
c[5]	-89
c[6]	0
c[7]	62
c[8]	-3
c[9]	1
c[10]	6453
c[11]	78

Numero di posizione dell'elemento nell'array c

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Array

- Gli elementi di un array sono come normali variabili

```
c[ 0 ] = 3;
printf( "%d", c[ 0 ] );
```

- Operazioni eseguite con un indice. If x equals 3

```
c[ 5 - 2 ] == c[ 3 ] == c[ x ]
```

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Definizione degli Array

- Nella definizione degli array, specificare

- Tipo dell'array

- Nome

- Numero di elementi

```
arrayType arrayName[ numberOfElements ];
```

- Esempi:

```
int c[ 10 ];
```

```
float myArray[ 3284 ];
```

- Definizione di array multipli dello stesso tipo

- Formato simile alle variabili regolari

- Esempio:

```
int b[ 100 ], x[ 27 ];
```

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Esempi con Array

- Inizializzazione

```
int n[ 5 ] = { 1, 2, 3, 4, 5 };
```

- Se non ci sono sufficienti valori di inizializzazione, gli elementi più a destra diventano 0

```
int n[ 5 ] = { 0 }
```

- Tutti gli elementi hanno il valore 0
- Se ci sono troppi valori di inizializzazione, viene prodotto un errore sintattico
- Gli array in C non hanno il controllo dei limiti

- Se la dimensione è omessa, gli inizializzatori la determinano

```
int n[ ] = { 1, 2, 3, 4, 5 };
```

- 5 inizializzatori, perciò array di 5 elementi

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```

1  /* Fig. 6.3: fig06_03.c
2  Initializing an array */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main()
7  {
8  int n[ 10 ]; /* n is an array of 10 integers */
9  int i;      /* counter */
10
11 /* initialize elements of array n to 0 */
12 for ( i = 0; i < 10; i++ ) {
13     n[ i ] = 0; /* set element at location i to 0 */
14 } /* end for */
15
16 printf( "%s\n", "Element", "Value" );
17
18 /* output contents of array n in tabular format */
19 for ( i = 0; i < 10; i++ ) {
20     printf( "%7d\n", i, n[ i ] );
21 } /* end for */
22
23 return 0; /* indicates successful termination */
24
25 } /* end main */

```





Outline

fig06_03.c

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

Element	Value
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0



 [Outline](#) 7
 **Program Output**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

```



1 /* Fig. 6.4: fig06_04.c
2   Initializing an array with an initializer list */
3 #include <stdio.h>
4
5 /* function main begins program execution */
6 int main()
7 {
8   /* use initializer list to initialize array n */
9   int n[ 10 ] = { 32, 27, 64, 18, 95, 14, 90, 70, 60, 37 };
10  int i; /* counter */
11
12  printf( "%s%13s\n", "Element", "Value" );
13
14  /* output contents of array in tabular format */
15  for ( i = 0; i < 10; i++ ) {
16    printf( "%7d%13d\n", i, n[ i ] );
17  } /* end for */
18
19  return 0; /* indicates successful termination */
20
21 } /* end main */

```

 [Outline](#) 8
 **fig06_04.c**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

Element	Value
0	32
1	27
2	64
3	18
4	95
5	14
6	90
7	70
8	60
9	37



 [Outline](#) 9
 **Program Output**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

```



1 /* Fig. 6.5: fig06_05.c
2  Initialize the elements of array s to the even integers from 2 to 20 */
3 #include <stdio.h>
4 #define SIZE 10
5
6 /* function main begins program execution */
7 int main()
8 {
9     /* symbolic constant SIZE can be used to specify array size */
10    int s[ SIZE ]; /* array s has 10 elements */
11    int j;        /* counter */
12
13    for ( j = 0; j < SIZE; j++ ) ( /* set the values */
14        s[ j ] = 2 + 2 * j;
15    ) /* end for */
16
17    printf( "%s\n", "Element", "Value" );
18
19    /* output contents of array s in tabular format */
20    for ( j = 0; j < SIZE; j++ ) {
21        printf( "%7d\n", s[ j ] );
22    } /* end for */
23
24    return 0; /* indicates successful termination */
25
26 } /* end main */

```

 [Outline](#) 10
 **fig06_05.c**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

Element	Value
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	16
8	18
9	20



 [Outline](#) 11
 **Program Output**

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

```

1 /* Fig. 6.6: fig06_06.c
2   Compute the sum of the elements of the array */
3 #include <stdio.h>
4 #define SIZE 12
5
6 /* function main begins program execution */
7 int main()
8 {
9   /* use initializer list to initialize array */
10  int a[ SIZE ] = { 1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45 };
11  int i;          /* counter */
12  int total = 0; /* sum of array */
13
14  /* sum contents of array a */
15  for ( i = 0; i < SIZE; i++ ) {
16    total += a[ i ];
17  } /* end for */
18
19  printf( "Total of array element values is %d\n", total );
20
21  return 0; /* indicates successful termination */
22
23 } /* end main */

```

 [Outline](#) 12
 **fig06_06.c**

Program Output

Total of array element values is 383


© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

```

1  /* Fig. 6.10: fig06_10.c
2     Treating character arrays as strings */
3  #include <stdio.h>
4
5  /* function main begins program execution */
6  int main()
7  {
8     char string1[ 20 ];           /* reserves 20 characters */
9     char string2[] = "string literal"; /* reserves 15 characters */
10    int i;                       /* counter */
11
12    /* read string from user into array string2 */
13    printf("Enter a string: ");
14    scanf( "%s", string1 );
15
16    /* output strings */
17    printf( "string1 is: %s\nstring2 is: %s\n"
18           "string1 with spaces between characters is: \n",
19           string1, string2 );
20
21    /* output characters until null character is reached */
22    for ( i = 0; string1[ i ] != '\0'; i++ ) {
23        printf( "%c ", string1[ i ] );
24    } /* end for */
25

```

13

 [Outline](#)




fig06_10.c (Part 1 of 2)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

14



Array come parametri a funzioni

- Passaggio di array
 - Per passare un array come argomento a una funzione, specificare il nome dell'array senza parentesi quadre


```
int myArray[ 24 ];
myFunction( myArray, 24 );
```

 - La dimensione dell'array è in genere passata come ulteriore parametro
 - Gli array sono passati per referenza
 - Il nome dell'array è l'indirizzo del suo primo elemento
 - La funzione conosce dove l'array è memorizzato
 - Vengono modificate le posizioni originali in memoria
- Passaggio di elementi singoli dell'array
 - Passaggio per valore
 - Passare il nome con l'indice (i.e., myArray[3]) alla funzione

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

Array come parametri a funzioni

- Prototipo di funzione

```
void modifyArray( int b[], int arraySize );
```

- I nomi dei parametri sono opzionali nel prototipo

- `int b[]` potrebbe essere scritto `int []`
- `int arraySize` potrebbe essere semplicemente `int`

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```

1  /* Fig. 6.13: fig06_13.c
2     Passing arrays and individual array elements to functions */
3  #include <stdio.h>
4  #define SIZE 5
5
6  /* function prototypes */
7  void modifyArray( int b[], int size );
8  void modifyElement( int e );
9
10 /* function main begins program execution */
11 int main()
12 {
13     int a[ SIZE ] = { 0, 1, 2, 3, 4 }; /* initialize a */
14     int i; /* counter */
15
16     printf( "Effects of passing entire array by reference:\n\nThe "
17            "values of the original array are:\n" );
18
19     /* output original array */
20     for ( i = 0; i < SIZE; i++ ) {
21         printf( "%3d", a[ i ] );
22     } /* end for */
23
24     printf( "\n" );
25

```



Outline

16

fig06_13.c (Part 1
of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.


```

26  /* pass array a to modifyArray by reference */
27  modifyArray( a, SIZE );
28
29  printf( "The values of the modified array are:\n" );
30
31  /* output modified array */
32  for ( i = 0; i < SIZE; i++ ) {
33      printf( "%3d", a[ i ] );
34  } /* end for */
35
36  /* output value of a[ 3 ] */
37  printf( "\n\nEffects of passing array element "
38          "by value:\n\nThe value of a[3] is %d\n", a[ 3 ] );
39
40  modifyElement( a[ 3 ] ); /* pass array element a[ 3 ] by value */
41
42  /* output value of a[ 3 ] */
43  printf( "The value of a[ 3 ] is %d\n", a[ 3 ] );
44
45  return 0; /* indicates successful termination */
46
47 } /* end main */
48

```



[Outline](#)

17

fig06_13.c (Part 2 of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

```

49 /* In function modifyArray, "b" points to the original array "a"
50    in memory */
51 void modifyArray( int b[], int size )
52 {
53     int j; /* counter */
54
55     /* multiply each array element by 2 */
56     for ( j = 0; j < size; j++ ) {
57         b[ j ] *= 2;
58     } /* end for */
59
60 } /* end function modifyArray */
61
62 /* In function modifyElement, "e" is a local copy of array element
63    a[ 3 ] passed from main */
64 void modifyElement( int e )
65 {
66     /* multiply parameter by 2 */
67     printf( "Value in modifyElement is %d\n", e * 2 );
68 } /* end function modifyElement */

```



[Outline](#)

18

fig06_13.c (Part 3 of 3)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

Effects of passing entire array by reference:

The values of the original array are:
0 1 2 3 4

The values of the modified array are:
0 2 4 6 8

Effects of passing array element by value:

The value of a[3] is 6
Value in modifyElement is 12
The value of a[3] is 6

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

19

Outline

Program Output

Esercizi: Calcolo della Media, Mediana e Moda usando array

- Media
- Mediana – numero al centro di una lista ordinata
 - 1, 2, 3, 4, 5
 - 3 è la mediana
- Moda – numero che occorre più spesso
 - 1, 1, 1, 2, 3, 3, 4, 5
 - 1 è la moda

(per la soluzione consultare il libro)

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

20

◀ ▶

Array Multidimensionali

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Array Multidimensionali

- Array multidimensionali
 - Tabelle con righe e colonne (m x n array)
 - Come le matrici: specificare le righe, poi le colonne

	Column 0	Column 1	Column 2	Column 3
Row 0	a[0][0]	a[0][1]	a[0][2]	a[0][3]
Row 1	a[1][0]	a[1][1]	a[1][2]	a[1][3]
Row 2	a[2][0]	a[2][1]	a[2][2]	a[2][3]

Nome array

Indice di riga

Indice di colonna

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Array Multidimensionali

- Inizializzazione

- `int b[2][2] = { { 1, 2 }, { 3, 4 } };`

1	2
3	4

- Inizializzatori raggruppati per righe tra parentesi graffe

- Se non sufficienti, gli elementi non specificati sono settati a zero

- `int b[2][2] = { { 1 }, { 3, 4 } };`

1	0
3	4

- Referenziazione degli elementi

- Specificare la riga, poi la colonna

- `printf("%d", b[0][1]);`

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



```

1 /* Fig. 6.21: fig06_21.c
2   Initializing multidimensional arrays */
3 #include <stdio.h>
4
5 void printArray( const int a[][ 3 ] ); /* function prototype */
6
7 /* function main begins program execution */
8 int main()
9 {
10  /* Initialize array1, array2, array3 */
11  int array1[ 2 ][ 3 ] = { { 1, 2, 3 }, { 4, 5, 6 } };
12  int array2[ 2 ][ 3 ] = { 1, 2, 3, 4, 5 };
13  int array3[ 2 ][ 3 ] = { { 1, 2 }, { 4 } };
14
15  printf( "Values in array1 by row are:\n" );
16  printArray( array1 );
17
18  printf( "Values in array2 by row are:\n" );
19  printArray( array2 );
20
21  printf( "Values in array3 by row are:\n" );
22  printArray( array3 );
23
24  return 0; /* Indicates successful termination */
25
26 } /* end main */
27

```

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.



Outline

fig06_21.c (Part 1 of 2)



```

28 /* function to output array with two rows and three columns */
29 void printArray( const int a[ [ 3 ] ] )
30 {
31     int i; /* counter */
32     int j; /* counter */
33
34     /* loop through rows */
35     for ( i = 0; i <= 1; i++ ) {
36
37         /* output column values */
38         for ( j = 0; j <= 2; j++ ) {
39             printf( "%d ", a[ i ][ j ] );
40         } /* end inner for */
41
42         printf( "\n" ); /* start new line of output */
43     } /* end outer for */
44
45 } /* end function printArray */

```

Values in array1 by row are:
1 2 3
4 5 6
Values in array2 by row are:
1 2 3
4 5 0
Values in array3 by row are:
1 2 0
4 0 0

25

 Outline

fig06_21.c (Part 2 of 2)

Program Output

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

26

Esercizi

- Scrivere un programma che utilizza due funzioni per caricare un vettore di numeri interi e visualizzarne il contenuto. Ripetere l'esercizio per le matrici.
- Scrivere un programma che utilizza due diverse funzioni per implementare l'algoritmo di "ricerca lineare" e quello di "ricerca binaria" in un vettore
- Scrivere un programma che richiede all'utente l'immissione del punteggio raggiunto da sei studenti, li memorizza nel vettore voti e ne determina il maggiore, il minore e la media.
- Calcolo del prodotto di due matrici

© Copyright 1992–2004 by Deitel & Associates, Inc. and Pearson Education Inc. All Rights Reserved.

