

CORSO DI PROGRAMMAZIONE
A.A. 2014-15

<h1>Dispensa 10</h1>

Laboratorio

Dott. Mirko Ravaioli
e-mail: mirko.ravaioli@unibo.it

<http://www.programmazione.info>

Array

Un array è un insieme di celle di archiviazione, ciascuna dello stesso tipo di dati e con lo stesso nome. Le celle di un array prendono il nome di elementi.

Un array si dice **monodimensionale** quando ha in solo indice di riferimento. In indice è un numero tra parentesi quadre che segue il nome di un array e identifica ciascun elemento. Un esempio: nel programma di gestione delle spese mensili si utilizzerebbe la riga che segue per dichiarare un array di tipo float:

```
float spese[12];
```

Tale array di nome *spese*, può contenere 12 elementi, ciascuno dei quali è assolutamente equivalente ad una variabile di tipo float. E' possibile creare array di qualsiasi tipo riconosciuto dal C. In C gli indici vengono numerati a partire da 0, per cui i 12 elementi di *spese* sono indicizzati da 0 a 11. Nell'esempio precedente le spese relative al mese di gennaio potrebbero essere memorizzate nell'elemento *spese[0]*, quello di febbraio in *spese[1]* e così via.

Quando viene dichiarato un array, il compilatore riserva un blocco di memoria abbastanza grande da contenerlo interamente. Gli elementi dell'array vengono memorizzati all'interno di locazioni di memoria contigue. Il punto in cui vengono dichiarati gli array all'interno del codice sorgente è molto importante. Così come accade per le variabili semplici, il punto dove avviene la dichiarazione influisce sull'ambito dell'array. Gli elementi dell'array possono essere utilizzati ovunque possa esserlo una variabile dello stesso tipo. Si accede ai singoli elementi riportando il nome dell'array seguito tra parentesi quadre. Ad esempio l'istruzione seguente memorizza il valore 78.16 all'interno del secondo elemento dell'array (bisogna ricordarsi che il primo elemento è *spese[0]* e non *spese[1]*):

```
spese[1] = 78.16;
```

Allo stesso modo, l'istruzione:

```
spese[10] = spese[7];
```

assegna all'elemento dell'array *spese[10]* una copia del valore contenuto nell'elemento *spese[7]*. Quando ci si riferisce ad un elemento di un array, l'indice può essere una costante letterale, come in questo esempio oppure un'espressione (compreso il valore di un altro elemento dell'array). Ecco alcuni esempi:

```
float spese[12];
int a[10];
...
spese[i] = 100; /* i è una variabile intera */
spese[2 + 3] = 100; /* equivale a spese[5] */
spese[a[2]] = 100; /* a[] è un array intero */
```

Considerando l'ultimo esempio, prendendo un array di interi denominato *a[]*, se all'interno dell'elemento *a[2]* è contenuto il valore 8 l'istruzione:

```
spese[a[2]] = 100;
```

ha lo stesso significato di quella seguente:

```
spese[8] = 100;
```

Quando si utilizzano gli array si tenga ben presente lo standard di numerazione: in un array di *n* elementi gli indici possibili vanno da 0 a *n* - 1. Se si cerca di utilizzare l'indice *n*, il programma dà errore.

Vediamo un esempio: programma che legga 5 valori interi inseriti da un utente e successivamente li stampi nello stesso ordine.

```
#include <stdio.h>

int main()
```

```
{
    int elenco[5];
    int i;;

    /*lettura dei valori*/
    for (i = 0; i < 5; i++)
    {
        printf("\n Inserire numero %d: ",i);
        scanf("%d",&elenco[i]);
    }

    /*stampa dei valori inseriti*/
    for (i = 0; i < 5; i++)
        printf("\n %d",elenco[i]);

    return 0;
}
```

Vediamo una variazione del programma precedente: la stampa dei valori inseriti deve essere in senso inverso.

```
#include <stdio.h>

int main()
{
    int elenco[5];
    int i;;

    /*lettura dei valori*/
    for (i = 0; i < 5; i++)
    {
        printf("\n Inserire numero %d: ",i);
        scanf("%d",&elenco[i]);
    }

    /*stampa dei valori in senso inverso*/
    for (i = 4; i >= 0; i--)
        printf("\n %d",elenco[i]);

    return 0;
}
```

Vediamo un altro un esempio: programma che permetta ad un utente di inserire 10 numeri e successivamente ne calcoli la somma.

```
#include <stdio.h>

int main()
{
    int elenco[10];
    int i, somma;

    /*lettura dei valori*/
    for (i = 0; i < 10; i++)
    {
        printf("\n Inserire numero %d: ",i);
```

```
        scanf("%d",&elenco[i]);
    }

    /*calcolo della somma*/
    somma = 0;
    for (i = 0; i < 10; i++)
        somma += elenco[i];

    printf("\n la somma e\' %d",somma);

    return 0;
}
```

Quando viene dichiarato un array, è possibile inizializzarlo in totalmente o in parte facendo seguire la dichiarazione da un segno di uguale e da un elenco di valori racchiusi tra parentesi graffe e separati da virgole. I valori in elenco vengono assegnati in ordine agli elementi dell'array a cominciare da quello con indice 0. Ad esempio l'istruzione seguente dichiara un array di dimensione 4 e associa all'elemento di indice 0 il valore 100, all'elemento di indice 1 il valore 200, all'elemento di indice 2 il valore 300 e all'elemento di indice 3 il valore 400:

```
int array[4] = {100, 200, 300, 400};
```

oppure

```
int array[] = {100, 200, 300, 400};
```

Nel secondo caso la dimensione viene omessa e il compilatore crea un array della dimensione sufficiente per contenere i valori con cui lo si inizializza.

E' comunque possibile specificare un numero di valori di inizializzazione inferiore alla capacità:

```
int array[10] = {1, 2, 3};
```

Scrivere un programma che permetta di cercare un numero all'interno del vettore. In particolar il programma deve restituire l'indice della cella in cui il valore cercato è memorizzato.

```
#include <stdio.h>

int main()
{
    int elenco[10] = {13,6,98,2,31,5,88,64,26,91};
    int i, cerca, trovato;

    printf("inserire il numero da cercare: ");
    scanf("%d",&cerca);

    trovato = 0;
    for (i = 0; i < 10; i++)
    {
        if (elenco[i] == cerca)
        {
            trovato = 1;
            break;
        }
    }

    if (trovato)
        printf("\n Il numero e\' nella cella %d",i);
    else
        printf("\n Il numero cercato non e\' stato trovato");
}
```

```
        return 0;
    }
```

Scrivere un programma che conti tutte le occorrenze di un numero all'interno di un vettore di interi.

```
#include <stdio.h>

int main()
{
    int elenco[10] = {13,6,98,2,31,5,88,64,26,91};
    int i, cerca, conta;

    printf("inserire il numero da cercare: ");
    scanf("%d",&cerca);

    conta = 0;
    for (i = 0; i < 10; i++)
    {
        if (elenco[i] == cerca)
            conta++;
    }
    printf("\n Il numero compare %d volte",conta);

    return 0;
}
```

Esercizio 1

Scrivere un programma che inverta l'ordine degli elementi di un array

Esercizio 2

Scrivere un programma che inverta di posto due elementi di un array

Esercizio 3

Scrivere un programma che confronti 2 array di interi.