

Array e sottoprogrammi

Dott.ssa Veronica Rossano
rossano@di.uniba.it
<http://www.di.uniba.it/~rossano>

Array e sottoprogrammi

- Piuttosto che passare singoli elementi dei vettori ad un sottoprogramma è possibile passare interi array come argomenti
- La dichiarazione di un sottoprogramma nella forma seguente

```
tipo_risultato nome_sottoprogramma (tipo_elem nome_array[ ] );  
■ void rileva_temperatura (double temp[ ] );
```

indica che il parametro formale conterrà solo l'indirizzo del primo elemento dell'array e che tutte le operazioni avranno come risultato la **manipolazione dell'array originale e non di una sua copia**

...Array e sottoprogrammi

- È da notare che:
 - La dichiarazione del parametro formale non indica il numero di elementi contenuti nel vettore
 - Il passaggio di parametri potrebbe avvenire allo stesso modo semplicemente utilizzando un puntatore al primo elemento dell' array

Esempio

- Definire un sottoprogramma che inizializzi tutti gli elementi di un qualsiasi vettore ad un valore specificato dall'utente

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
```

**Dichiarazione di un
vettore come parametro
formale**

```
void inizializza(int vettore[], int dim, int valore);
void visualizza(int vettore[], int dim);
```

```
int
main (void)
{
    int n, val_inizializzazione;
    int vett[MAX];
    printf("Inserire il numero degli elementi del vettore (Massimo 100) -->");
    scanf("%d", &n);
    printf("Inserire il valore di inizializzazione degli elementi del vettore -->");
    scanf("%d", &val_inizializzazione);
    inizializza(vett, n, val_inizializzazione);
    visualizza(vett, n);
    system("pause");
    return(0);
}
```

**Passaggio di un vettore
come parametro reale**

```
void inizializza(int vettore[], int dim, int valore)
{
    int i;
    for (i=0; i<dim; i++)
    {
        vettore[i]=valore;
    }
}
```

Inizializzazione di un
vettore

```
void visualizza(int vettore[], int dim)
{
    int i;
    for (i=0; i<dim; i++)
    {
        printf("Il %d valore e' --> %d\n", (i+1), vettore[i]);
    }
}
```

Visualizzazione degli
elementi di un vettore

Esercizio

- Scrivere un programma che restituisca il più grande elemento in un vettore qualsiasi dato in input

Massimo

Algoritmo

- Inserire il numero di valori
- Leggere il primo
- Porre il massimo al primo
- Mentre non si sono letti tutti gli elementi
 - Leggere il successivo
 - Se è maggiore del massimo
 - Porre il massimo a questo numero
- Comunicare il massimo


```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100

void inizializza(int vettore[], int dim);
int valore_massimo(int vettore[], int dim);

int
main (void)
{
    int n, massimo;
    int vett[MAX];
    printf("Inserire il numero degli elementi del vettore (Massimo 100) -->");
    scanf("%d", &n);
    inizializza(vett, n);
    massimo=valore_massimo(vett, n);
    printf("L'elemento piu' grande del vettore e' --> %d \n", massimo);
    system("pause");
    return(0);
}
```

Inizializzazione di un vettore

Calcolo del massimo elemento di un vettore

```
void inizializza(int vettore[], int dim)
{
    int i;
    for (i=0; i<dim; i++)
    {
        printf("\nInserire il %d valore --> ", (i+1));
        scanf("%d", &vettore[i]);
    }
}
```

Inizializzazione di un
vettore con input
dall'utente

```
int valore_massimo(int vettore[], int dim)
{
    int i, massimo_corrente;
    massimo_corrente=vettore[0];
    for (i=1; i<dim; i++)
    {
        if (vettore[i]>massimo_corrente)
            massimo_corrente=vettore[i];
    }
    return(massimo_corrente);
}
```

Calcolo del massimo
elemento in un vettore

Vettori come parametri di input

- È possibile indicare nell'intestazione di una funzione se un array deve essere considerato solo di input, ogni tentativo di modifica ai singoli elementi sarà segnalato come errore sintattico dal compilatore
- La sintassi

```
Tipo_ris nome_funzione (const tipo_elem nome_array[ ]);  
■ void inizializza (const int vettore[ ], int dim);  
■ void visualizza (const int vettore[ ], int dim);
```

Array multidimensionali e sottoprogrammi

- Il passaggio di array multidimensionali come parametri di un sottoprogramma ricalca quanto già detto per gli array monodimensionali con il vincolo che l'unica dimensione che può essere lasciata non definita è la prima

```
Tipo_risultato nome_funzione (tipo_elem nome_array[ ] [dim2]);
```

- ```
void rileva_temperatura (double temp[][mesi]);
```

Chiamata di un sottoprogramma

```
rileva_temperatura (temperatura);
```