

Informatica 1

Corso di Laurea Triennale in Matematica

Gianluca Rossi

`gianluca.rossi@uniroma2.it`

Dipartimento di Matematica
Università di Roma "Tor Vergata"

6: Array



- Gli *array* permettono di creare *vettori* o *matrici* di elementi della stesso tipo;
- In C un array ad una dimensione è un vettore. Contiene un numero di elementi fissato. Tale numero è la *lunghezza* dell'array.
- Un array a k dimensioni è una matrice con k indici.
- Un array di dimensione k può esser visto come un array monodimensionale i cui elementi sono array di dimensione $k - 1$.



Array di dimensione 1

```
tipo nomearray[lunghezza];
```

Crea un array di dimensione 1 contenente *lunghezza* elementi ognuno dei quali è di tipo *tipo*.

lunghezza deve essere un valore numerico.

```
nomearray[i];
```

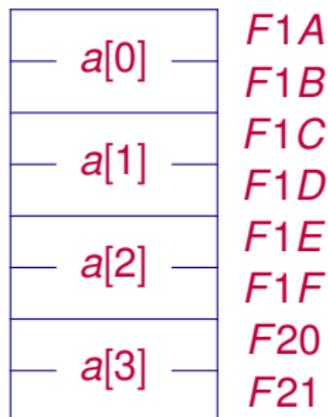
con $0 \leq i < \textit{lunghezza}$ si riferisce all' $i + 1$ esimo elemento del vettore (nomearray[0]; è il primo elemento del vettore).



Array monodimensionali (2)

- Supponendo che il tipo *int* occupi due parole di memoria:

```
int a[4];
```



Array monodimensionali (3)

```
#include <stdio.h>
```

```
main(){
```

```
    int a[10], i;
```

```
    a[0] = 1;
```

```
    for(i = 1; i < 10; i++)
```

```
        a[i] = a[i-1] * i;
```

```
    for (i = 0; i < 10; i++)
```

```
        printf("a [%d] = %d\n", i, a[i]);
```

```
}
```

- Cosa contiene `a[i]` al termine della computazione?



Array monodimensionali (4)

- ... oppure facendo uso di una “costante”

```
#include <stdio.h>
#define N 10

main(){
    int a[N], i;

    a[0] = 1;
    for(i = 1; i < N; i++)
        a[i] = a[i-1] * i;

    for (i = 0; i < N; i++)
        printf("a [%d] = %d\n", i, a[i]);
}
```



Array monodimensionali (5)

- ... oppure utilizzando il **valore** di una variabile

```
#include <stdio.h>
```

```
main(){  
    int n = 10;  
    int a[n], i;  
  
    a[0] = 1;  
    for(i = 1; i < n; i++)  
        a[i] = a[i-1] * i;  
  
    for (i = 0; i < n; i++)  
        printf("a [%d] = %d\n", i, a[i]);  
}
```



Array monodimensionali (6)

- Nel caso in cui per la dimensione di un vettore venga utilizzato il valore di una variabile n , a questa gli deve essere assegnato il giusto valore prima della definizione del vettore;

Giusto `int n = 10;`
`int a[n];`

Sbagliato `int n ;`
`int a[n];`
`n = 10;`

- Se il valore di n cambia dopo la definizione del vettore, la dimensione del vettore resta comunque invariata.

```
int n = 10;  
int a[n];  
n = 15;  
a[12] = 4; /* a[12] non esiste!!! */
```



- Definizioni ed inizializzazioni alternative.

```
int a[4] = { 1, 2, 3, 4};  
int b[] = { 5, 6, 7};    /* dimensione implicita */  
int c[3] = { 1 };       /* c[0] = 1, c[1] = c[2] = 0 */  
int d[6];               /* d[0] ... d[5] e' indefinito */
```



```
#include <stdio.h>
main(){
    int n = 10, i;
    printf("Dimensione del vettore? "); scanf("%d",&n);
    int a[n];
    /* la vera dim. del vettore*/
    printf("|a| = %d\n", sizeof(a)/sizeof(int));
    for(i = 0; i < n; i++){
        printf("a[%d] = ? ", i); scanf("%d", &(a[i]));
    }
    for(i=0; i < n; i++)
        printf("a[%d] = %d\n", i, a[i]);
}
```

- Qual è il numero di elementi del vettore *a*?

