

Array frastagliati

Violetta Lonati

Università degli studi di Milano
Dipartimento di Informatica

Laboratorio di algoritmi e strutture dati
Corso di laurea in Informatica

Argomenti

Array frastagliati

Array di stringhe

Argomenti da linea di comando

Array di stringhe

Un vettore bidimensionale di caratteri può essere pensato come un array di stringhe di lunghezza costante: ogni riga contiene una stringa della stessa lunghezza.

```
char colori1[6][10] = {"rosso", "blu", "oltremare"  
                      "verde", "nero", "giallo"};
```

- ▶ Il numero di colonne è pari almeno alla lunghezza massima delle stringhe (incluso il fine stringa): in questo esempio occupo lo spazio per $6 \times 10 = 60$ caratteri.
- ▶ Buona parte di questo spazio è occupata dal carattere nullo `'\0'`.
- ▶ NB: se si usa un vettore bidimensionale come parametro di funzione, è necessario specificare la seconda dimensione, ad esempio:

```
void stampa( char c[][10] );
```

Array di stringhe - 2

Il modo più corretto (ed efficiente) di gestire un insieme di stringhe di lunghezza variabile è attraverso l'uso di un array **frastagliato**, ovvero di un array di puntatori a char:

```
char *colori2[6] = {"rosso", "blu", "oltremare"  
                   "verde", "nero", "giallo"};
```

- ▶ In questo caso occupo lo spazio di 6 puntatori a char, più lo spazio strettamente necessario a contenere le 6 stringhe, pari a $(5 + 1) + (3 + 1) + (9 + 1) + (5 + 1) + (4 + 1) + (6 + 1) = 38$ caratteri.
- ▶ Il nome di un array di stringhe può essere considerato come un puntatore a stringa. Ad esempio, come parametro di funzione si può usare indifferentemente: `char *c[]` oppure `char **c`.
- ▶ NB: nei parametri di funzione devo usare dichiarazioni diverse se uso array frastagliati o array bidimensionali:
`char c[][10]` non equivale a `char **c!!`

Argomenti da linea di comando

Ogni programma C può avere degli argomenti passati da linea di comando. Per poterli usare è necessario che la funzione `main` sia definita con due parametri, chiamati solitamente `argc` e `argv`.

```
int main ( int argc, char *argv[] ) { ... }
```

oppure

```
int main ( int argc, char **argv ) { ... }
```

- ▶ `argc` è pari al numero degli argomenti (incluso il nome del comando);
- ▶ `argv` è un array frastagliato di stringhe, di lunghezza `argc+1`:
 - ▶ `argv[0]` è il nome del comando;
 - ▶ `argv[1]` è il primo argomento;
 - ▶ `argv[argc-1]` è l'ultimo argomento;
 - ▶ `argv[argc]` è il puntatore `NULL` (...);

Argomenti da linea di comando - esempio

```
/* somma.c - programma che somma i suoi argomenti */
#include <stdio.h>
#include <stdlib.h>

int main( int argc, char *argv[] ) {
    int i, somma = 0;
    for ( i = 1; i < argc; i++ )
        somma += atoi( argv[i] );

    printf( "Somma = %d\n", somma );
    return 0;
}
```

```
$ gcc -o somma somma.c
```

```
$ ./somma 1 3 10
```

```
Somma = 14
```