

A decorative vertical bar on the left side of the slide, consisting of several vertical lines of varying shades of blue and grey. To the right of these lines are several blue circles of different sizes, arranged in a vertical sequence. The largest circle is at the top, and the number '1' is centered within a medium-sized circle below it.

Laboratorio di Programmazione 1

Docente: dr. Damiano Macedonio
Lezione 17 - 24/03/2014

Stringhe di Caratteri

Inizializzazione

- Una stringa è una qualsiasi sequenza di caratteri racchiusa tra apici doppi.
 - `"a"`: è una stringa.
 - `'a'`: non è una stringa, è un carattere (tipo `char`).
- Per gestire variabili che contengono una stringa è necessario ricorrere agli array di caratteri.
 - `char word[] = {'H', 'e', 'l', 'l', 'o'};`
 - Array di dimensione 5 determinata dal compilatore.
- Se si utilizza un array di dimensione prefissata non è detto che la dimensione dell'array coincida con la lunghezza della stringa in esso contenuta.
 - `char word[10]`
 - `word[0] = 'a';`
 - `word[1] = 'b';`
 - L'array ha dimensione 10 ma la stringa ha lunghezza 2.

Stringhe di Caratteri

Inizializzazione

- Si utilizza il carattere nullo `'\0'` per segnalare la fine di una stringa.
 - Evita di mantenere un contatore che indica la fine della stringa quando questa è memorizzata in un array di dimensione fissa.
 - `char word[] = {'H', 'e', 'l', 'l', 'o', '\0'};`
- È possibile inizializzare un array di caratteri specificando direttamente una stringa, anziché una lista di singoli caratteri.
 - `char word[] = {"Hello"};`
 - Notare che viene aggiunto automaticamente il carattere nullo in fondo all'array. Quindi `word` ha dimensione 6.
 - Se si specifica direttamente la dimensione dell'array è necessario lasciare lo spazio per il carattere nullo finale.
 - `char word[6] = {"Hello"};`
 - È anche possibile omettere le parentesi graffe:
 - `char word[] = "Hello";`

Stringhe di Caratteri

Operazioni

- Si può accedere ai singoli caratteri che compongono una stringa con l'usuale notazione degli array.
 - `char word[] = "Hello";`
 - `word[2] → 'l'`
 - `Word[1] = 3 → "H3llo"`
- Per determinare se due stringhe di caratteri sono uguali *non* si può utilizzare l'operatore `==` è necessario confrontare esplicitamente le due stringhe carattere per carattere.
 - Usando un ciclo `while` che scorre entrambe finché non si raggiunge il carattere nullo in una delle due, oppure il carattere corrente nelle due stringhe è diverso.

Stringhe di Caratteri

Formattazione

- La funzione `printf` può essere utilizzata con formato speciale `%s` per visualizzare un array di caratteri *che termina con il carattere nullo*.
 - `printf("%s\n", word)`
- La funzione `scanf` può essere utilizzata con il formato speciale `%s` per leggere una stringa di caratteri finché non incontra: uno *spazio vuoto*, un *carattere di tabulazione*, o la *fine linea*.
 - `char word[81];`
 - `scanf("%s", word);`
 - **Notare che in questo caso non c'è il carattere `&` prima del nome della variabile.**
 - Se l'input è "Hello world." viene letto e assegnato a `word` solo "Hello".
 - Aggiunge automaticamente il carattere nullo al termine della stringa letta.

Stringhe di Caratteri

Formattazione

- Se l'utente digita più caratteri della dimensione dell'array senza uno spazio, un carattere di tabulazione o un fine linea, la funzione `scanf` eccede la capacità dell'array.
- È possibile indicare un numero massimo di caratteri letti dalla `scanf`, indipendentemente dal numero di caratteri digitati, inserendo tale numero tra i simboli `%` e `s` nella stringa di formato.
 - `char word[81];`
 - `scanf("%80s", word)`
 - Notare che l'array ha dimensione 81, mentre il valore massimo specificato è 80 per lasciare spazio al carattere nullo finale automaticamente aggiunto da `scanf`.