Le stringhe

Le stringhe

- Il termine “stringa” indica una sequenza di caratteri atta a rappresentare una frase
  - Un nome, un testo, un codice ed in generale una sequenza di simboli per il trattamento di testi
- Le stringhe non costituiscono un tipo *primitivo* del C++ e di conseguenza non sono ammesse come operandi dalla maggior parte degli operatori (compreso l'operatore di assegnazione)
- Sono tuttavia riconosciute da alcuni operatori (come per esempio gli operatori di flusso di I/O del C++) e da numerose funzioni di libreria del C che hanno il compito specifico di manipolare le stringhe
  - Concatenazione di stringhe
  - Confronto tra stringhe
  - Calcolo della lunghezza
  - …

Le stringhe in C++

- In C++, una stringa viene definita come un **array di caratteri che termina con il carattere NULL**, specificato come ‘\0’ e avente valore nullo
- In memoria, le stringhe sono degli array di tipo char, con una particolarità in più, che le fa riconoscere da operatori e funzioni come stringhe e non come normali array:
  - l'elemento dell'array che segue l'ultimo carattere della stringa deve contenere il carattere NULL (detto in questo caso terminatore);
  - si dice pertanto che una stringa é un "*array di tipo char null terminated*".

Le stringhe in C++

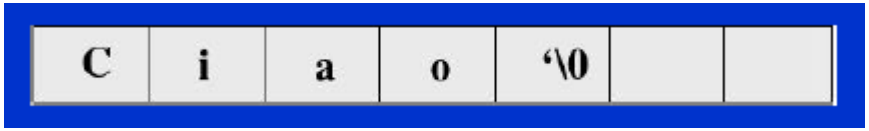
- A causa del valore NULL, è necessario che un array di caratteri venga dichiarato più grande di un carattere rispetto alla stringa più lunga che deve contenere
- Per esempio per dichiarare una stringa **str che dovrà** contenere al massimo 10 caratteri:

char str[11];

- in questo caso si lascia lo spazio per il terminatore NULL alla fine della stringa

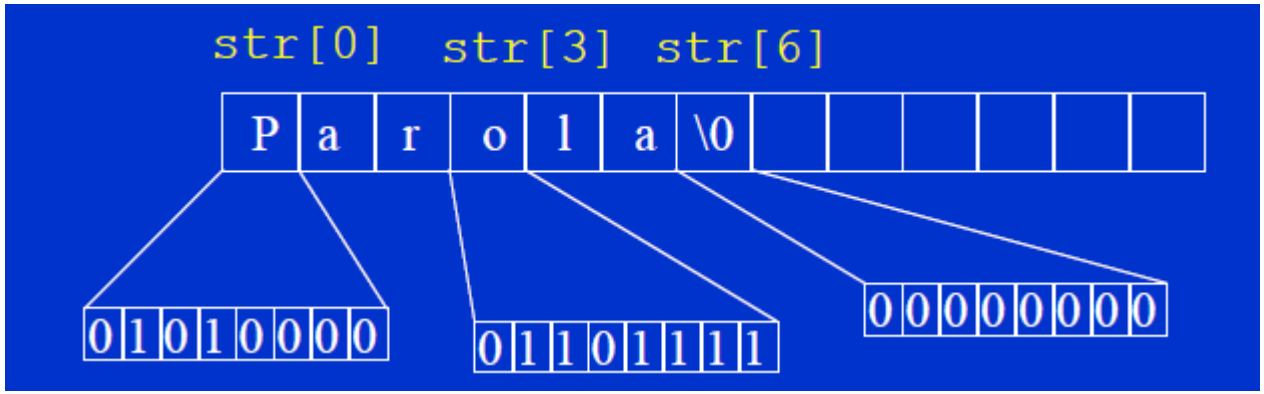
Le stringhe in C++: inizializzazione

- char Saluto[] = "Ciao";
- Equivale a:
- char Saluto[] = { 'C', 'i', 'a', 'o', '\0' };
- ➔ sizeof(Saluto)=5 e non 4, per via del carattere di terminazione
  - Nel caso 1, viene aggiunto automaticamente
  - Nel caso 2, è aggiunto esplicitamente
- Altro esempio:
- char Saluto[7] = “Ciao”;



Rappresentazione di stringhe di caratteri

- char str[13] = “Parola”;



- La stringa è terminata dal carattere “nullo” (‘\0’)
- Gli algoritmi di manipolazione terminano quando trovano tale carattere (informazione *tappo)*

Rappresentazione di stringhe di caratteri

```
char stringa1[] = "parola1";
```

- I seguenti sono esempi di semplici array di caratteri (non stringhe):

```
char Minnie[ ] = { 'M', 'i', 'n', 'n', 'i', 'e' };
char Pluto[5] = { 'P', 'l', 'u', 't', 'o' };
```

- In questi casi però non si ottiene una stringa (sequenza di caratteri terminata da '\0‘), ma semplici array di caratteri il cui numero di elementi è esattamente quello specificato

Lettura di stringhe

- **cin>>stringa;**
  - termina la lettura di una stringa non appena incontra un carattere di spaziatura (spazi, tab e new line)
- **gets(stringa)** ;
  - legge una linea dallo **stdin** fino alla fine della riga, aggiungendo il carattere tappo '\0‘
  - Non viene eseguito nessun controllo sulla dimensione della stringa
- **cin.getline(stringa, NMAX);**
  - Legge una linea dallo **stdin** fino alla fine della riga, aggiungendo il carattere tappo '\0’, oppure fino al raggiungimento del numero di elementi NMAX

Comandi di input per le stringhe

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

int main()
{
 const int DIM_VETT=30;
 char parola[DIM_VETT];

 cin >> parola;                    // utilizzando cin
 //gets(parola);                   // utilizzando gets
 //cin.getline(parola, DIM_VETT);  // utilizzando cin.getline
 cout << "Hai inserito " << parola << endl;

 system("PAUSE");
 return 0;
}
```

Alcune funzioni di libreria per le stringhe

- Libreria <string.h>
- strcpy(a, da) : per copiare il contenuto della stringa da a quella a
- strcat(s1, s2) : aggiunge s1 a s2
- strcomp(s1, s2) : confronta le stringhe s1 e s2, restituendo 0 se sono uguali
- strlen(s) : restituisce la lunghezza della stringa s