**14° lezione prog1**

**LE STRINGHE**

Le stringhe sono array di caratteri  che sono memorizzati in memoria come celle adiacenti.

Immagine che contiene testo, Carattere, numero, linea

Descrizione generata automaticamente

1. **la prima inizializzazione è corretta** perché si ha alla fine il carattere 0
2. **la seconda inizializzazione è corretta**  in quanto si è inserito il carattere del finestrino(\0)
3. **la terza inizializzazione è errata p**erché non ha il carattere di fine stringa (se non si utilizza la notazione (\0 o 0) le funzioni che si utilizzano per le stringhe non identificano la fine di questa)
4. **char s[]=”Ciao”;**In questo modo il carattere di fine stringa viene messo in automatico dal compilatore. Con questa dichiarazione sto allocando un vettore di carattere nello stack  Pertanto si denunce che è modificabile     (cioè si possono modificare vari caratteri (es s[2]=1;))
5. **char \*ps=”Ciao1”;** in questo modo viene allocato un Array di char  di sei  caratteri più il carattere di fine stringa. In questo modo si sta andando a conservare un puntatore ad una zona di memoria lo stack read only. In questo caso \*ps andrebbe dichiarato costante(const  char\*ps=”ciao”;).Se si cerca di modificare il puntatore si avrà un errore a tempo di esecuzione.

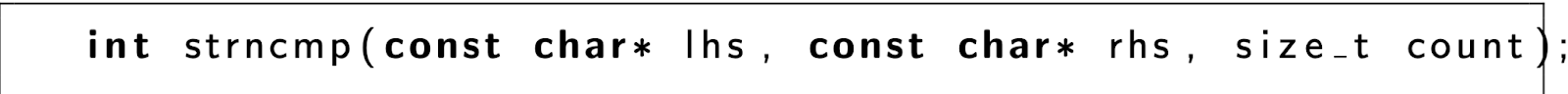
**Funzioni della libreria standard per le stringhe**

Per includere queste funzioni si utilizza l’header string.h, Ci sono alcune funzioni che modificano le stringa e altre no(const char). Quando si deve andare a modificare la stringa non bisogna dichiarare la costante invece quando non si vuole che venga modificata va dichiarata  costante. Es const  char\*ps=”ciao”;

 Si hanno svariate funzioni di questa libreria:

* **char \*strcpy(char ∗ destination , const char ∗ source );**Si utilizza per copiare una stringa in un'altra.In sostanza copia la stringa Source nell'area di memoria puntata del Destination e non fa nessun controllo sulla dimensione dell'area di memoria di Destination.
* **int strcmp(const char \*lhs, cost char \*rhs);**  Si utilizza per fare un controllo lessicografico di due stringhe o array di caratteri. Si vanno a controllare i singoli caratteri(questo avviene  con la tabella ASCII). Restituisce un intero.
* Se le stringhe sono identiche **restituisce 0**
* se la prima stringa  è maggiore della seconda restituisce **un intero  maggiore di zero**
* Se invece la  seconda stringa  è minore della prima restituisce **un intero minore di 0**

Ci sono anche le funzioni di copia e di confronto lessicografico in modalità length-bounded. limita il confronto i primi count caratteri.





Si anche un altra funzione che si utilizza la ricerca di sotto stringhe strstr()

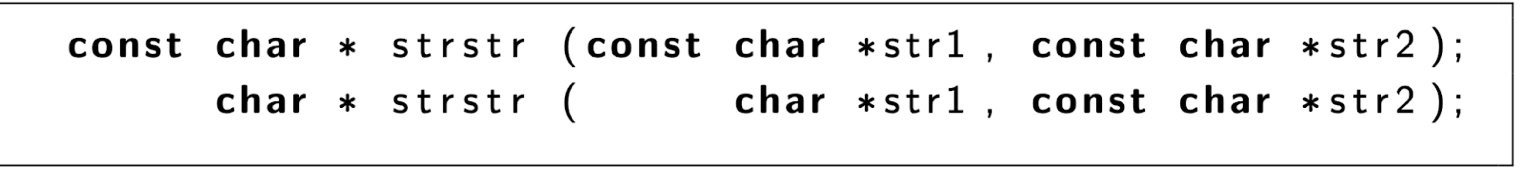


Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

Questa funzione restituisce un puntatore al primo carattere della prima  occorrenza di str-2 trovate in str-1 .

Si hanno delle funzioni che permettono di convertire  una stringa in numeri:

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

Si utilizzano con l’header stdlib.h convertono solo i numeri non le lettere. Es “11Ciao” restituirebbe 11.

Se la stringa  non inizia con un numero restituisce 0.

La stessa cosa si ha quando si deve convertire una stringa in numeri virgola mobile. con le funzioni

* atoi()
* atol()
* atoll()

**famiglia di funzioni strtoX**

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Questa funzione restituisce il numero riconosciuto nella stringa.Si va ad passare anche una cella che contiene il puntatore al primo carattere non usato nella conversione Per controllare se nessuna conversione sia avvenuta bisogna controllare che \*endptr=nptr;

**sscanf() ed fprintf()**

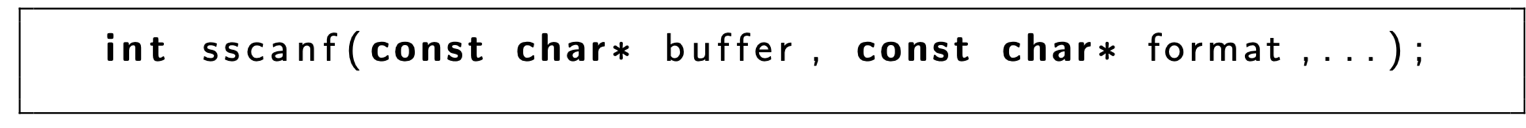
* **La funzione sscanf()** fa parte della famiglia delle funzioni scanf() esegue turtte le funzioni della scanf() ma con la differenza che fa il parsing della stringa passata come primo argomento (un puntatore a carattere).   
  Ha il seguente prototipo:   
  

Immagine che contiene testo, Carattere, schermata, bianco

Descrizione generata automaticamente

Il codice di cui sopra ha un comportamento simile alla funzione atoi()

* La funzione fprintf() fa parte della famiglia di funzioni printf().

Ha il seguente prototipo:



Si utilizza per scrivere nel buffer la stringa specificata nel parametro format.

Immagine che contiene testo, ricevuta, Carattere, bianco

Descrizione generata automaticamente

**Funzioni che operano con i singoli caratteri con l’heder ctye.h**

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamenteImmagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

Sono funzioni che permettono di capire quale carattere si ha ad esempio un numero.

**Funzioni toupper() e tolower()**

Queste funzioni si utilizzano per convertire lettere maiuscole in minuscole o viceversa.

* **toupper():** restituisce un intero e vuole come parametro un intero che rappresenta il carattere da convertire. Tale funzioni si utilizza per convertire lettre maiuscole in minuscole,
* **tolower();**restituisce un intero e vuole come parametro un intero che rappresenta il carattere da convertire. Tale funzioni si utilizza per convertire lettre minuscola in maiuscola,