LE STRINGHE

Le stringhe

Le stringhe sono array monodimensionali di tipo char che terminano con \0.

Un vettore di caratteri non è necessariamente una stringa perché non sempre termina con \0.

\0 è un carattere di controllo che indica al compilatore la fine della stringa.
Si usano per rappresentare parole, frasi, codici alfanumerici e più in generale qualunque tipo di sequenza numerica non aritmetizzabile.

Le stringhe

Una stringa si può definire come vettore di caratteri

char v [5]; ← dichiara una stringa di 5 caratteri

char v[]="ciao"; ← dichiara una stringa e la inizializza a ciao. Lo spazio in memoria viene preparato opportunamente per accogliere la stringa.

Le stringhe

char v='A'; ← dichiara un carattere singolo

A

char v[]="A"; ←dichiara una stringa di un solo carattere

A	\0

← **Viene inserito automaticamente e non è visualizzabile con la printf.**

Una stringa occupa una posizione di memoria in più rispetto ad un vettore di caratteri che contiene la stessa parola.

Elementi di una stringa

Per raggiungere i singoli caratteri che compongono una stringa si può lavorare come con i vettori, precisando la posizione dell'elemento desiderato e ricordando che in C il primo elemento di un vettore si trova in posizione 0.

v[i]

↑

Indice (numero o espressione intera positiva)

← v[2]

Dati singoli (caratteri) che possono essere usati in assegnazioni

Elementi di una stringa

Per raggiungere i singoli caratteri che compongono una stringa si può lavorare come con i vettori, precisando la posizione dell'elemento desiderato e ricordando che in C il primo elemento di un vettore si trova in posizione 0.

F
A
M
E
\0

← v[2]='R'

F
A
R
E
\0

← v[0]='M'

M
A
R
E
\0

Stringhe e puntatori

IL NOME DI UN VETTORE
(e quindi anche di una stringa)
E' UN PUNTATORE ALLA SUA PRIMA POSIZIONE.

Una stringa può essere dichiarata anche come

char *a;

E posso usare le operazioni aritmetiche sul puntatore per scorrere gli elementi della stringa.

Assegnazione valori a stringa

Per assegnare i valori ad una stringa è possibile:

- Assegnazione diretta

```
char v[4]="ape";
char *v="ape";
```

a
p
e
\0

~~char v[4]={'a', 'p', 'e'};~~

~~char v[4];~~
~~v="ape";~~

~~char v[];~~

Assegnazione valori a stringa

Per assegnare i valori ad una stringa è possibile:

- Assegnazione diretta
- Acquisizione da tastiera

Serve per leggere anche frasi con spazi bianchi.

```
char a[100];
gets(a);
```

```
char a[100];
scanf("%[^\n]s", a);
```

```
char a[100];
scanf("%s", a);
```

La & non serve perché a è già un indirizzo di memoria.

Visualizzazione del contenuto di una stringa

Per visualizzare il contenuto di una stringa è sufficiente una singola istruzione di stampa:

```
printf("%s", a);
```

```
puts(a);
```

Gestione delle stringhe

Per la gestione delle stringhe il linguaggio C mette a disposizione una libreria apposita:

string.h

Alcune utili funzioni della libreria sono:

strcat
strcmp
strcpy
strlen

Gestione delle stringhe

strcat(stringa1, stringa2)

Riceve due stringhe e le concatena (unisce) mettendo il risultato nella prima dei due parametri passati.

```
char a[100]="Il mare è ", *b="blu";
strcat(a,b);
puts(a);
```

Il mare è blu

Gestione delle stringhe

strcmp(stringa1, stringa2)

Riceve due stringhe e le confronta.

Restituisce un intero:

0 se le stringhe sono uguali

>0 se la prima stringa è maggiore

<0 se la seconda stringa è maggiore

L'ordinamento è da intendersi in senso lessicografico.

Gestione delle stringhe

strcmp(stringa1, stringa2)

```
char a[]="albero", *b="altalena";
int x;
x=strcmp(a,b);
printf("%d",x);
```

-1

```
char a[100]="giallo", *b="giallo";
strcmp(a,b);
x=strcmp(a,b);
printf("%d",x);
```

0

Gestione delle stringhe

strcpy(stringa1, stringa2)

Riceve due stringhe e copia il contenuto della stringa2 nella stringa 1 (fino al \0).

Il primo contenuto di stringa1 viene perso.

```
char a[100]="giallo", *b="blu";
strcpy(a,b);
puts(a);
```

blu

Accertarsi che in stringa1 vi sia spazio sufficiente ad accogliere stringa2.

Gestione delle stringhe

strcpy(stringa1, stringa2)

La funzione strcpy è utile per assegnare i valori ad una stringa successivamente alla dichiarazione.

```
char a[100];
a="blu";
puts(a);
```

```
char a[100];
strcpy(a, "blu");
puts(a);
```

blu

Gestione delle stringhe

strlen(stringa1)

Restituisce un intero, pari alla lunghezza della stringa passata come argomento, compresi spazi bianchi, al netto del carattere di controllo \0.

```
char a[100]="Leggo un libro giallo";
int x;
x=strlen(a);
printf("%d",x);
```

21