Le stringhe

Prof. Orazio Mirabella

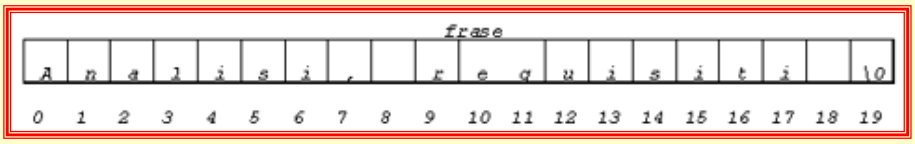
Generalità sulle stringhe 1/2

- Una variabile di tipo char consente di memorizzare un singolo carattere
- E' comodo poter trattare come una sola unità un insieme di caratteri alfanumerici, detto *stringa*;
- a questo scopo si possono utilizzare gli array di char.
- Es: char a[10];

Oppure

- char frase[] = "Analisi, requisiti ";

  In tal caso, il numero di elementi è determinato dalla quantità di caratteri presenti tra doppi apici più uno, il carattere *null* (\0) che chiude la stringa



Generalità sulle stringhe 2/2

Quale è la differenza tra le due inizializzazioni?:

- char d = 'r';
- char b[] = "r";
- La prima assegna alla variabile d di tipo char il valore r, la seconda assegna all'array b[] la sequenza di caratteri r e \0; in quest'ultimo caso si tratta effettivamente di una stringa

- Il carattere terminatore \0 ci permette di trattare le stringhe senza conoscere a priori la dimensione.

Esempio 1

```
/* Visualizzazione caratteri di una stringa */
#include <stdio.h>
char frase[] = "Analisi, requisiti ";
main()
{
int i=0;
while(frase[i]!='\0') {
printf("%c = %d = %o \n", frase[i], frase[i], frase[i]);
i++;
}
}
```

è possibile usare l'istruzione printf tramite la specifica del formato %s: printf("%s", frase);
Tale istruzione, se inserita nel Listato di sopra , restituirebbe: Analisi, requisiti

```
A = 65 = 101
n = 110 = 156
a = 97 = 141
l = 108 = 154
i = 105 = 151
s = 115 = 163
i = 105 = 151
, = 44 = 54
 = 32 = 40
r = 114 = 162
e = 101 = 145
q = 113 = 161
u = 117 = 165
i = 105 = 151
s = 115 = 163
i = 105 = 151
t = 116 = 164
i = 105 = 151
 = 32 = 40
```

Esempio 2

```
/* Copia di una stringa su un'altra */
#include <stdio.h>
char frase[] = "Analisi, requisiti ";
main()
{
int i;
char discorso[80];
for(i=0; (discorso[i]=frase[i])!='\0'; i++)
;
printf(" originale: %s \n copia: %s \n", frase, discorso);
}
```

Variante

```
for(i=0; frase[i]!='\0'; i++)
discorso[i]=frase[i];
discorso[i]='\0';
```

- anche il carattere terminatore viene copiato nell'array discorso, che sarà così correttamente delimitato.

```
originale: Analisi, requisiti
copia: Analisi, requisiti
```

Esempio 3

```
/* Concatenazione di due stringhe */
#include <stdio.h>
char frase[160] = "Analisi, requisiti ";
main()
{
char dimmi[80];
int i, j;
printf("Inserisci una parola: ");
scanf("%s", dimmi);
for(i=0; (frase[i])!='\0'; i++)
;
for(j=0; (frase[i]=dimmi[j])!='\0'; i++,j++)
;
printf("frase: %s \n", frase);
}
```

Libreria string.h

- La funzione strcpy consente di copiare stringa2 su stringa1:
  strcpy(stringa1, stringa2);
- La funzione strncpy permette invece di copiare i primi *n* caratteri di stringa2 in stringa1:
  strncpy(stringa1, stringa2, n);
- La funzione strcat consente di concatenare stringa2 a stringa1:
  strcat(stringa1, stringa2);
- La funzione strcmp serve a confrontare stringa2 con stringa1:
  strcmp(stringa1, stringa2);
  Se risultano essere uguali viene restituito zero, se stringa1 è maggiore di stringa2 viene restituito un valore positivo, altrimenti un valore negativo.

Esempio

```
/* Confronto tra due stringhe con strcmp */
#include <stdio.h>
#include <string.h>
char prima[160] = "mareggiata";
main()
{
char seconda[80];
int i, x;
printf("Inserisci una parola: ");
for(i=0; ((seconda[i]=getchar())!='\n') && (i<80); i++)
;
seconda[i] = '\0';
if( (x = (strcmp(prima, seconda))) == 0)
printf("Sono uguali\n");
else
if(x>0)
printf("la prima è maggiore della seconda\n");
else
printf("la seconda è maggiore della prima\n");
}
```