

Gestione delle stringhe in C

Generalità

- Una stringa è un vettore di caratteri il cui ultimo elemento è un carattere terminatore (o di fine stringa), codificato dal carattere di codice 0 e rappresentato in C dal carattere '\0'.
- Il vettore di caratteri che rappresenta la stringa sarà perciò formato da un numero di elementi pari al numero di caratteri della stringa più uno (il carattere di fine stringa).
- La stringa "Rossi" è rappresentata in C da un vettore di 6 caratteri: 5 caratteri per memorizzare **Rossi** più il carattere di terminazione:

0	1	2	3	4	5
R	o	s	s	i	\0

Dichiarazione di una stringa

- In C per dichiarare una stringa in grado di contenere N caratteri occorre dichiarare un vettore di N+1 caratteri.
- Es. per dichiarare una variabile stringa atta a contenere un codice fiscale occorre dichiarare un vettore di 17 caratteri:
 - 16 caratteri per memorizzare i caratteri e le cifre del codice fiscale
 - 1 carattere per il terminatore di stringa ('\0')

```
char codice_fiscale[17];
```

Lettura e scrittura di una stringa

- L'identificatore di formato utilizzato dalle funzioni di lettura (`scanf`) e scrittura (`printf`) per manipolare le stringhe è il `%s`.

•Esempio: Leggere e stampare una stringa

```
main(){
    char s[20];
    printf("Inserisci una stringa: ");
    scanf("%s", s);
    printf("Hai inserito la stringa: %s", s);
}
```

Come si vede si è scritto `scanf("%s", s)` e non `scanf("%s", &s)`.
Il motivo di ciò sarà chiarito quando sarà introdotto il concetto di puntatore.

Funzioni di Libreria

- La libreria standard del C mette a disposizione diverse funzioni per la gestione delle stringhe. Per utilizzare tali funzioni è necessario includere nel proprio file sorgente `string.h`.

```
#include <string.h>
```

Funzione strcmp

- La funzione `strcmp` consente di confrontare due stringhe passate come parametro.
- Date due stringhe, `stringa1` e `stringa2`,

$$\text{strcmp}(\text{stringa1}, \text{stringa2}) = \begin{cases} 0 & \text{se } \text{stringa1} == \text{stringa2} \\ <0 & \text{se } \text{stringa1} < \text{stringa2} \\ >0 & \text{se } \text{stringa1} > \text{stringa2}. \end{cases}$$

Scrivere un programma che legga e confronti due stringhe

```
#include <stdio.h>
#include <string.h>

main()
{
    char stringa1[20], stringa2[20];

    printf("inserisci la prima stringa: ");
    scanf("%s", stringa1);
    printf("inserisci la seconda stringa: ");
    scanf("%s", stringa2);

    if (strcmp(stringa1, stringa2)== 0)
        printf("le due stringhe sono uguali");
    else if (strcmp(stringa1, stringa2) < 0)
        printf("%s < %s", stringa1, stringa2);
    else
        printf("%s > %s", stringa1, stringa2);
}
```

Funzione strcpy

- Poichè le stringhe sono vettori di caratteri, non è possibile assegnare ad una variabile stringa un'altra variabile stringa utilizzando l'operatore =.
- Le possibili soluzioni sono due:
 - copiare elemento per elemento dalla stringa sorgente alla stringa destinazione fino a quando si incontra il carattere di fine stringa;
 - utilizzare la funzione strcpy .
- Il formato della funzione strcpy è il seguente:
`strcpy (destinazione, sorgente)`
- dove destinazione è la variabile stringa di destinazione, sorgente è la variabile stringa sorgente.
- La stringa destinazione deve essere grande da ospitare tutti gli elementi della stringa sorgente.

Scrivere un programma che letta una stringa immessa da tastiera la copia in un'altra stringa.

```
#include <stdio.h>
#include <string.h>

main()
{
    char sorgente[20], destinazione[20];

    printf("inserisci una stringa: ");
    scanf("%s", sorgente);
    strcpy(destinazione, sorgente);
    printf("destinazione = %s", destinazione);
}
```

Funzione strlen

- La funzione `strlen(stringa)` restituisce la lunghezza della stringa passata come parametro.
- Per lunghezza della stringa si intende il numero di caratteri contenuti nella stringa escluso il carattere di terminazione.

Es. Scrivere un programma che chieda all'utente di inserire una parola e ne restituisca la lunghezza.

```
#include <stdio.h>
#include <string.h>
main()
{
    char stringa[50];
    int lunghezza;

    printf("Scrivi una parola: ");
    scanf("%s", stringa);
    lunghezza = strlen(stringa);
    printf("%s è formata da %d caratteri.", stringa, lunghezza);
}
```

Funzione strcat

- La funzione `strcat(stringa1, stringa2)` consente di concatenare le due stringhe passate come parametro.
- Il risultato della concatenazione è memorizzato nel primo parametro.
- Se per esempio `stringa1` contiene "Giuseppe" e `stringa2` contiene "Rossi", allora a seguito dell'esecuzione di `strcat(stringa1, stringa2)` `stringa1` conterrà la stringa "GiuseppeRossi".

Scrivere un programma che letti il nome e cognome di un utente li concateni in un'unica stringa.

```
#include <stdio.h>
#include <string.h>

main()
{ char nome[10], cognome[10], nome_cognome[20];
  printf("Inserisci il tuo nome: ");
  scanf("%s", nome);
  printf("Inserisci il tuo cognome: ");
  scanf("%s", cognome);
  strcpy(nome_cognome, nome);
  strcat(nome_cognome, cognome);
  printf("nome+cognome = %s", nome_cognome);
}
```

Vettore di stringhe

- La dichiarazione di un vettore i cui elementi sono delle stringhe è analoga a quella dei vettori di elementi di tipo semplice C
- La sintassi da utilizzare per la dichiarazione di una variabile vettore di N elementi di tipo stringa di D caratteri è la seguente:

```
char NomeVettore[N][D];
```

Leggere e visualizzare un vettore di 10 stringhe di 20 caratteri

```
#include <stdio.h>
#define N 10
#define D 20

main()
{ char V[N][D];
  int i;

  for(i=0;i<N;i++)
  { printf("Inserisci una nuova stringa");
    scanf("%s",V[i]);
  }

  for(i=0;i<N;i++)
    printf("%s",V[i]);
}
```

Leggere un vettore di 10 stringhe di 20 caratteri e realizzare la ricerca sequenziale

```
#include <stdio.h>
#define N 10
#define D 20

main()
{ char V[N][D], stringa[D];
  int i, trovato;

  for(i=0; i<N; i++)
  { printf("Inserisci una nuova stringa");
    scanf("%s", V[i]);
  }
  printf("Inserire una stringa");
  scanf("%s", stringa);

  i=0; trovato=0;
  while(i<N && !trovato)
    if(strcmp(stringa, V[i])==0) trovato=1;
    else i++;

  if(trovato) printf("%s", V[i]);
  else       printf("Non e' presente");
}
```

Leggere un vettore di 20 stringhe di 30 caratteri e applicare il bubble sort

```
#include <stdio.h>
#include <string.h>
#define N 20
#define D 30

main()
{ char V[N][D], stringa[D];
  int i, sup;

  for(i=0; i<N; i++)
  { printf("Inserisci una nuova stringa");
    scanf("%s", V[i]);
  }
  for(sup=N-1; sup>0; sup--)
    for(i=0; i<sup; i++)
      if(strcmp(S[ind+1], S[ind])<0)
        { strcpy(aux, S[ind+1]);   strcpy(S[ind+1], S[ind]);
          strcpy(S[ind], aux);
        }
  for(i=0; i<N; i++)
    printf("%s\n", S[i]);
}
```