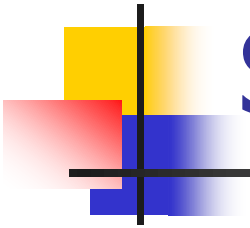


Stringhe

Concetti chiave

- Variabili strutturate: array di caratteri
- Carattere di fine stringa
- Copiare, concatenare, confrontare e convertire stringhe
- Funzioni standard `strcpy`, `strcat`, `strcat`, `strcmp`, `atoi`, `atof`
- Lettura stringa

Stringhe

- Per trattare insiemi di caratteri alfanumerici, *stringhe:* **array di `char`**.

```
char a[10];
char a[15] = "Automobile";
char frase[] = "Analisi, requisiti ";
```

- `frase` ha un numero di elementi uguale ai caratteri presenti tra doppi apici più uno, il carattere *null* (’`\0`’) che chiude la stringa.

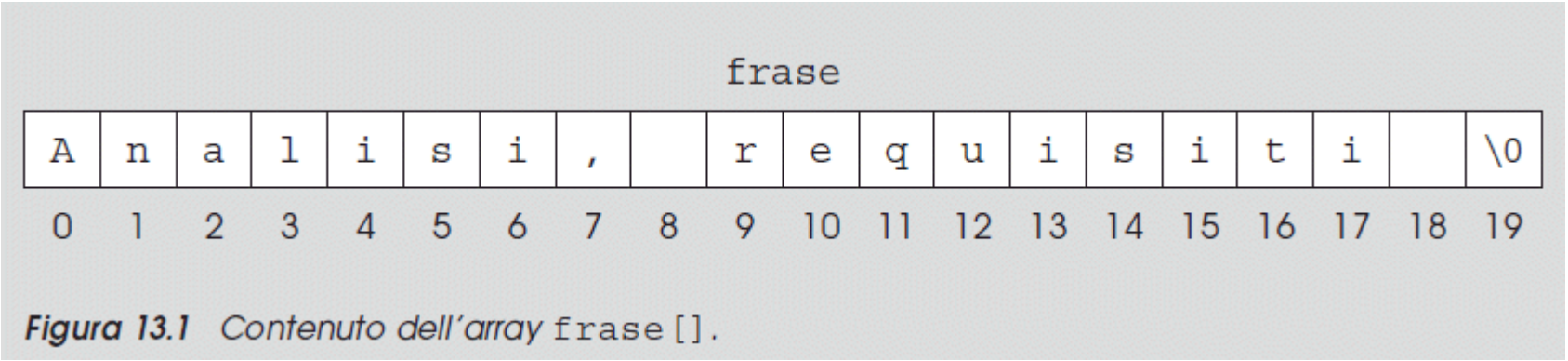


Figura 13.1 *Contenuto dell’array* `frase[]`.

Stringhe

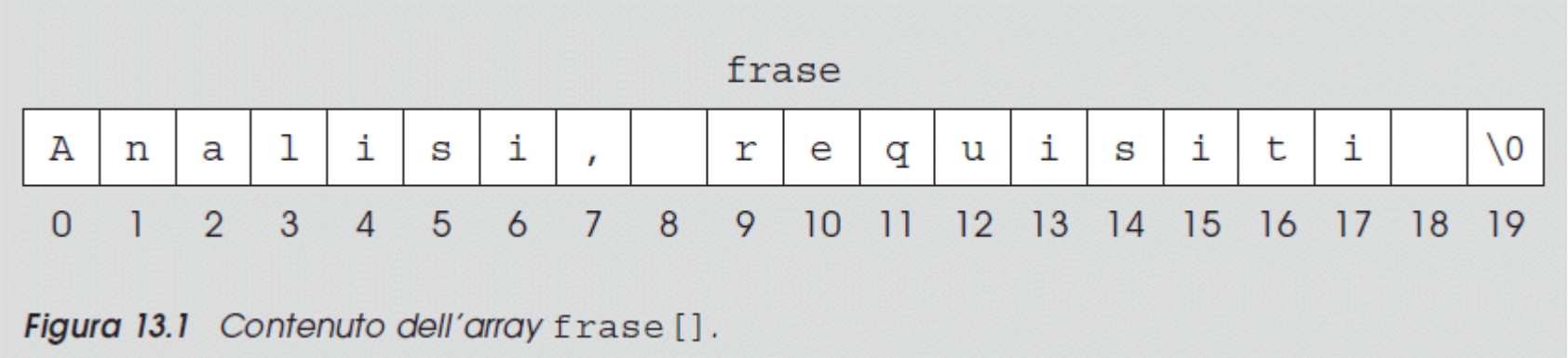


Figura 13.1 *Contenuto dell'array* `frase[]`.

Per visualizzazione un'intera stringa si usa la specifica del formato `%s`:

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

che restituisce: `Analisi, requisiti`

Stampa carattere per carattere la stringa e si blocca al carattere '`\0`'.

Stringhe

```
#include <stdio.h>

char frase[] = "Analisi, requisiti ";
int i;

int main(void)
{
    i=0;
    while(frase[i]!='\0') {
        printf("\n%c = %d ",frase[i],frase[i]);
        i++;
     }
}
```

Restituisce:

```
A = 65
n = 110
a = 97
l = 108
i = 105
s = 115
i = 105
, = 44
  = 32
r = 114
e = 101
q = 113
u = 117
i = 105
s = 115
i = 105
t = 116
i = 105
  = 32
```

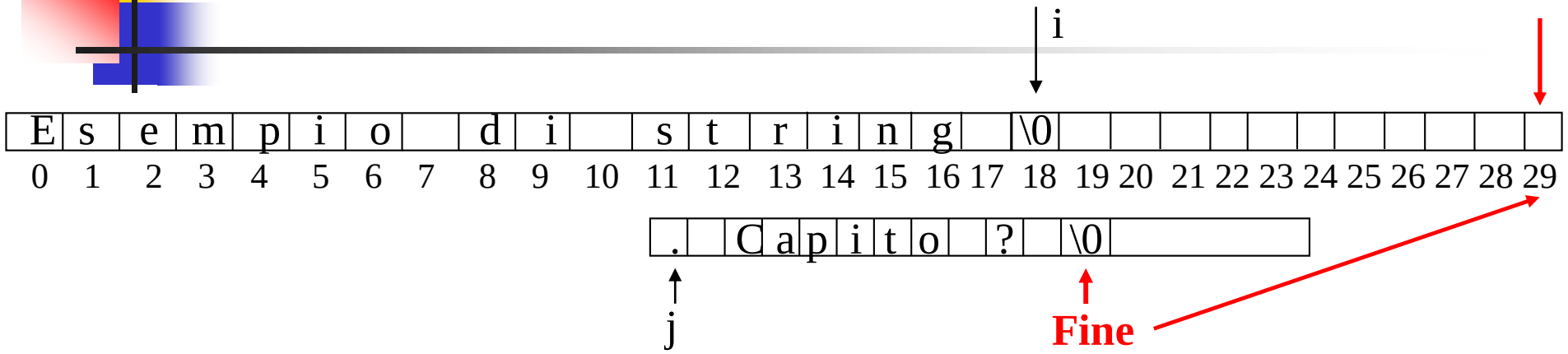
spazio

Stringhe - Copia

```
#include <stdio.h>

char s[20] = "Analisi, requisiti ", d[20];
int i;


int main(void)
{
   i=0;
   do
      d[i]=s[i];
   while (d[i++]!='\0');
   printf("\nOriginale: %s \nCopia: %s \n", s, d);
}
```

```
#include <stdio.h>

char s[20] = "Analisi, requisiti ", d[20];
int i;


int main(void)
{
   for(i=0; (d[i]=s[i])!='\0'; i++);
   printf("\nOriginale: %s \nCopia: %s \n", s, d);
}
```

Stringhe - Concatenazione



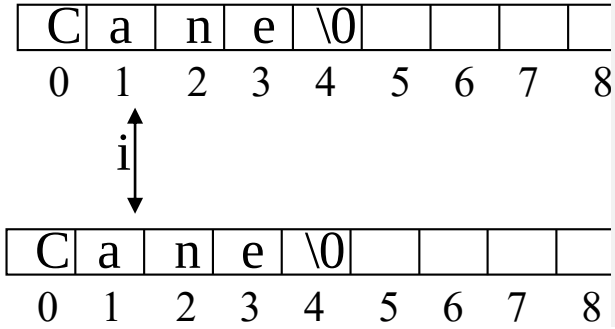

```
#include <stdio.h>
char frase[160] = "Esempio di string ";
char s[80] = ". Capito ? ";
int i, j;

int main(void)
{
   for(i=0; frase[i]!='\0'; i++);
   for(j=0; (frase[i]=s[j])!='\0'; i++, j++);
   printf("\nLa frase concatenata : %s ", frase);
   printf("\nLa sua lunghezza e' = %u \n", i);
}
```

29

Stringhe - Lunghezza

```
#include <stdio.h>
#define N 80

char frase[N]="Calcolo Lunghezza ";
unsigned int count;

int main(void)
{
    count=0;
    while (frase[count]!='\0') count++;
    printf("\nLunghezza di %s e' %u \n",frase,count);
}
```

Stringhe - Confronto




```
#include <stdio.h>
#define N 80

char s1[N]="Cane", s2[N]="Canale";
unsigned int i;

int main(void)
{
  for (i=0;s1[i]==s2[i];i++)
                    if (s1[i]=='\0') break;

  if (s1[i]-s2[i]>0)
     printf("\nLa stringa %s e' > della stringa %s ",s1,s2);
  else if (s1[i]-s2[i]<0)
            printf("\nLa stringa %s e' < della stringa %s ",s1,s2);
       else printf("\nLa stringa %s e' uguale alla stringa %s ",s1,s2);
}
```

Stringhe-Libreria string.h

Libreria string.h

- strcpy copia stringa2 su stringa1:

  strcpy(stringa1, stringa2);

- strcat concatena stringa2 a stringa1:

  strcat(stringa1, stringa2);

- strcmp confrontare stringa2 con stringa1:

  strcmp(stringa1, stringa2);

- strlen calcola la lunghezza di una stringa:

  strlen(stringa);

- Se sono uguali **zero**,
- se stringa1 > di stringa2 **valore positivo**,
- altrimenti un **valore negativo**.

Stringhe-Libreria stdlib.h

Libreria stdlib

- **i = atoi(stringa);**

converte una stringa in un intero.

- **d = atof(string);**

converte una stringa in un valore in virgola mobile **double**.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define N 80

char s1[N];
int i;
double val;

int main(void){
    strcpy(s1, "98993489");
    i = atoi(s1);
    val = atof(s1);
    printf("\nValore della Stringa %s = %d", s1, i);
    printf("\nValore della Stringa %s = %lf", s1, atof(s1));
}
```

Stringhe-Input

Come si leggono le stringhe ?

- scanf con il formato %s: **scanf("%s ",frase); SENZA & !!!!**
  - ✓si ferma anche quando incontra uno spazio; digitando "degli utenti" memorizza solo "degli".
  - ✓permette di leggere più caratteri di quanti ne abbia la stringa !!!!!!
- per leggere tutta la stringa, compresi spazi, eccetto gli a-capo:

  scanf("%[^\n]s", frase);
- Però rimane il problema che permette di leggere più caratteri di quanti ne abbia la stringa !!!!!!
- Prima dello standard C11, esisteva la funzione gets(), che permetteva di leggere stringhe compresi spazi.
  - ✓ E' stata soppressa.
  - ✓ In ogni caso non riusciva a capire se si superava la dimensione massima della stringa

Stringhe-Input

Esempio di scanf

Ricordarsi che: la lunghezza max di una stringa è = N-1 perché c'è il carattere '\0' da memorizzare

```
#include <stdio.h>
#include <string.h>
#define N 8
char s[N];

int main(void)
{
   printf("\nInserisci una stringa ");
   scanf("%[^\n]s",s);
   printf("\nLa stringa inserita e': %s ", s);
   printf("\ned ha lunghezza %u ",strlen(s));
}
```

Stringhe-Input

Esiste un'altra possibilità:

- fgets (stringa, lunghezza, stdin)
- Legge tutti i caratteri fino allo '\n' (incluso) e aggiunge '\n' se il numero di caratteri < lunghezza-1
- oppure ne legge esattamente (lunghezza-1) e aggiunge '\0'
- Problema: se la stringa inserita ha meno caratteri della lunghezza data, viene inserito alla fine il carattere '\n' e dopo '\0' !

Stringhe-Input

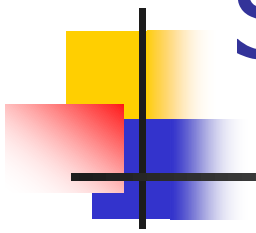
Esempio di fgets():

```
#include <stdio.h>

#define N 8
char s[N];

int main(void)
{
   printf("\n\nUSO FGETS che gestisce gli spazi e la lunghezza max");
   printf("\nInserisci una stringa di MAX %u caratteri ",N);
   fgets(s,N,stdin);
   printf("\nLa stringa inserita e': %s ", s);
   printf("\ned ha lunghezza %u ", strlen(s));
}
```

Miglioramento di fgets():

```
fgets(s,N,stdin);
if (strlen(s)<N-1) s[strlen(s) - 1] = '\0';
```

Stringhe-Input



…….meglio farsi un programma ad hoc di lettura stringhe…..!!!

Stringhe

Procedura perfetta per leggere qualunque stringa senza errori

```
#include <stdio.h>
#include <string.h>
#define N 25
#define FFLUSH while (getchar()!='\n')
char s[N];
int i;

int main(void){
  printf("\nInserisci una stringa di lunghezza max %u   ", N-1);
  for (i=0; i<N-1;i++)
      if ((s[i]=getchar())=='\n') break;
  if (i==N-1) FFLUSH;
  s[i]='\0';

  printf("\nLa stringa inserita e': %s ",s);
  printf("\ned ha lunghezza %u ",strlen(s));
}
```

Frase digitata

A	n	a	l	i	s	i	,		r	e	q	u	i	s	i	t	i		\n
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Frase copiata nel vettore s

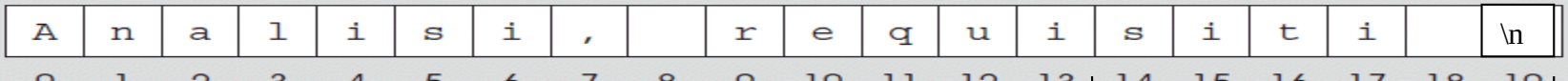
A	n	a	l	i	s	i	,		r	e	q	u	i	s	i	t	i		\0			
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22

Stringhe-Input

Procedura perfetta per leggere qualunque stringa senza errori

```
#define N 15
#define FFLUSH while (getchar()!='\n')

for (i=0; i<N-1;i++)
        if ((s[i]=getchar())=='\n') break;
if (i==N-1) FFLUSH;
s[i]='\0';

printf("\nLa stringa inserita e': %s ",s);
printf("\ned ha lunghezza %u ",strlen(s));
```

Frase digitata



Frase copiata nel vettore s

Il buffer viene pulito