



UNIVERSITÀ
DEGLI STUDI DELLA
TUSCIA

INFORMATICA

Programma

Dott. Franco Liberati
franco.liberati@unitus.it

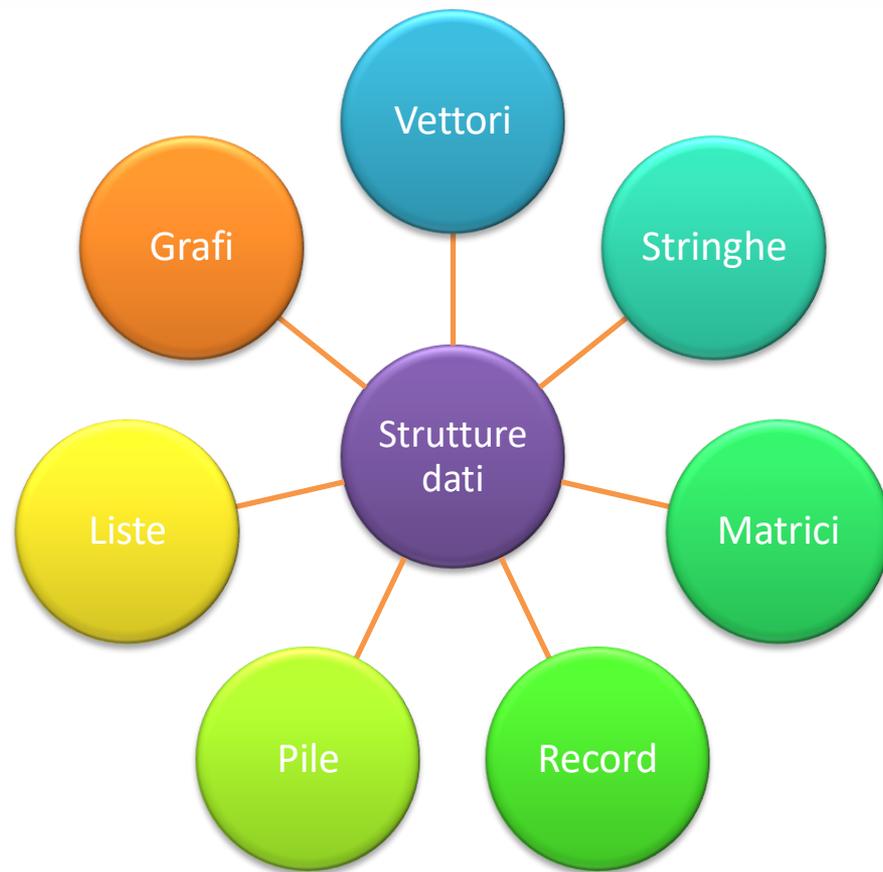


PROGRAMMA

DATI STRUTTURATI

Definizione. I **dati strutturati** (o strutture di dati) sono dati ottenuti mediante composizione di altri dati (di tipo semplice, oppure strutturato)

- Vettori (o array)
- Stringhe
- Matrici
- Liste
- Pile
- Grafi
- Record



PROGRAMMA

STRINGA

- ❑ Una **stringa** (*string*) è un vettore di caratteri con **terminatore**, cioè il valore zero (`\0`)

NB: IN ASCII

0-9 48-58

A-Z 65-90

a-z 97-122

Indice	Dato	Carattere
0	84	T
1	117	u
2	115	s
3	99	c
4	105	i
5	97	a
6	0	\0

PROGRAMMA

STRINGA

La sintassi per definire una **stringa** è:

```
char nome_variabile[dim];
```

☐ Dove

- ❖ **char** è il tipo del dato character
- ❖ *Nome_variabile* è un identificatore che specifica il vettore
- ❖ **Dim** è il numero di elementi che compongono il vettore (deve essere un valore intero positivo)

Nel linguaggio C il primo elemento ha indice 0

ESEMPIO

```
char stringa1[5];
```

//Definisce una stringa di 4 caratteri

#define dimensione 5

```
char stringa1[dimensione];
```

//Definisce una stringa di 4 caratteri

// si può usare dimensione all'interno del

//programma

// #define non richiede il ; dopo il valore

PROGRAMMA

STRINGA

La sintassi per definire e inizializzare un **vettore** è:

```
char nome_variabile[dim]={'chr1','chr2','chr3',...};
```

oppure

```
char nome_variabile[dim]="chr1chr2chr3...";
```

Per **prelevare un dato dal vettore** bisogna specificare l'indice

Per **sovrascrivere un dato di un vettore** bisogna specificare l'indice (in C il primo elemento del vettore è 0)

ESEMPIO

```
char stringa1[7]={'T','u','s','c','i','a','\0'};
char carattere;

carattere =stringa1[3];
// in carattere c'è il carattere 'c'

stringa1[1]='o';
// il terzo carattere della stringa muta in 'o':
// stringa1= "Toscia"
```

ESEMPIO

```
char stringa2[7]= "Toscia";
char carattere;

carattere =stringa2[3];
// in carattere c'è il carattere 'c'

stringa2[1]='o';
// il terzo carattere della stringa muta in 'o':
// stringa2= "Toscia"
```

PROGRAMMA

STRINGA lettura e scrittura



La sintassi per **leggere da tastiera** un vettore è:

```
char stringa[200];  
scanf ("%s", stringa);
```

La sintassi per **stampare a video** un vettore è:

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

La stampa riproduce su terminale i caratteri dall'inizio fino al terminatore

ESEMPIO

```
int main()  
{  
    char stringa[20];  
    printf("Inserisci una stringa: ");  
    scanf("%s", stringa);  
    printf("Hai inserito la stringa: %s", stringa);  
  
    return 0;  
}
```

PROGRAMMA

STRINGA: definizione

- ❑ La dimensione di una stringa ci dice quanti caratteri sono ammessi. Eccesso di quel valore (meno uno per inserire il terminatore `\0`) i restanti caratteri sono persi
- ❑ Nel caso in cui si decida di inizializzare la stringa con un minor numero di caratteri, degli elementi successivi non è noto il valore
- ❑ La perdita dello `\0` comporta la non possibilità di determinare la fine della stringa (condizione necessaria per la stampa con la funzione `scanf`)

```
char stringa[7];
```

```
scanf("%s", s);
```

```
// inserisco ciao sono 4 caratteri e gli altri 3?
```

Indice	Dato	Carattere
0	99	c
1	105	i
2	97	a
3	111	o
4	0	\0
5	?	?
6	?	?

IMPORTANTE: SIZE OF =7 LENGHT=4



PROGRAMMA

STRINGA errore comune 1

- ❑ Dopo la sua dichiarazione la stringa va trattata come un vettore a tutti gli effetti
- ❑ Non è possibile copiare una stringa in un'altra stringa con un'assegnazione

```
char s1[10] = "testo 1";  
char s1[10] = "testo 2";  
s1 = s2 //ERRORE
```

In questo caso deve essere copiato elemento per elemento

```
int i=0;  
while(s2[i]!='\0') {  
    s1[i] = s2[i];  
    i=i+1;  
}
```



PROGRAMMA

STRINGA errore comune 2

- ❑ Dopo la sua dichiarazione la stringa va trattata come un vettore a tutti gli effetti.
- ❑ Non è possibile confrontare una stringa con un'altra stringa con un operatore logico

```
bool test;
```

```
char s1[10] = "testo 1";
```

```
char s1[10] = "testo 1";
```

```
test= (s1 == s2) //ERRORE
```

In questo caso deve essere confrontato elemento per elemento

```
int i=0;
```

```
bool diverso=false;
```

```
while(s2[i]!='\0') {
```

```
    if (s1[i] != s2[i]) {diverso=true;}
```

```
    i=i+1;
```

```
}
```



PROGRAMMA

STRINGHE

Definire una stringa di 255 caratteri, leggere il contenuto da tastiera e calcolare la lunghezza (cioè il numero di caratteri)

```
#include <stdio.h>
int main() {
    char s1[255];
    int cont=0; int i=0;
    printf(" \n Inserisci la stringa: ");   scanf("%s", s1);
    while (s1[i]!=0) {
        i=i+1;
        cont=cont+1;
    }
    printf("La lunghezza della stringa è: %d", cont);
    return 0;
}
```



PROGRAMMA

STRINGHE

Leggere due stringhe di dimensione 10 e stampare se sono uguali o diverse

```
#include <stdio.h>
int main() {
    char s1[10]; char s2[10];
    printf("Inserisci la stringa n. 1\n"); scanf("%s", s1);
    printf("Inserisci la stringa n. 2\n"); scanf("%s", s2);
    int uguale = 1; //ipotizza che s1 e s2 siano uguali
    //verifica se s1 e s2 sono uguali
    int i = 0;
    for(i=0; s1[i]!='\0'; i++) {
        if (s1[i]!=s2[i]) { uguale = 0; }
    }
    //se è arrivato sul carattere terminatore di s1, verifica che sia così anche per s2
    if (s1[i]='\0' && s2[i]!='\0'){uguale = 0;} //s1 e s2 sono diverse

    if (uguale==1) {printf("Le due stringhe sono uguali");}
    else {printf("Le due stringhe sono diverse");}
    return 0;
}
```



PROGRAMMA

STRINGHE

Definire due stringhe di almeno 10 caratteri e concatenarle, cioè:

```
S1= "Paolino"
```

```
S2= "Paperino"
```

```
S1 ∪ S2= "PaolinoPaperino"
```

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

```
int main() {
```

```
    char s1[11]="Paolino";  char s2[11]="Paperino";  char s3[21];
```

```
    int i=0;int j=0;
```

```
    while (s1[i]!=0) {
```

```
        s3[i]=s1[i];
```

```
        i=i+1;
```

```
    }
```

```
    j=i;
```

```
    i=0;
```

```
    while (s2[i]!=0) {
```

```
        s3[j]=s2[i];
```

```
        i=i+1;
```

```
        j=j+1;
```

```
    }
```

```
    printf("La stringa %s concatenata alla stringa %s è: %s", s1,s2,s3);
```

```
    return 0;
```

```
}
```



Librerie

PROGRAMMA

LIBRERIA STRING.H: generalità

- ❑ La libreria **string.h** contiene una ricca serie di funzioni per operare sulle stringhe

Esempi:

- ❖ copiare una stringa in un'altra (strcpy)
- ❖ concatenare due stringhe (strcat)
- ❖ confrontare due stringhe (strcmp)
- ❖ cercare un carattere in una stringa (strchr)
- ❖ cercare una stringa in un'altra (strstr)

- ❑ Per usare la libreria è necessario includerla nel codice

```
#include "string.h"
```

PROGRAMMA

LIBRERIA STRING.H: LUGHEZZA

- ❑ La funzione restituisce la lunghezza della stringa strstr, senza contare il terminatore
- ❑ La sintassi è:

```
int strlen(char* str);
```

```
#include <stdio.h>
#include <string.h>
int main(){
    char s1[]="Paperino";
    char s2[50]="Minnie";
    printf("\n%d",sizeof(s1));           //Stampa 9
    printf("\n%d",strlen(s1));          //Stampa 8
    printf("\n%d",sizeof(s2));         //Stampa 50
    printf("\n%d",strlen(s2));         //Stampa 6
    return 0;
}
```

PROGRAMMA

LIBRERIA STRING.H: COMPARAZIONE

- ❑ La funzione restituisce un valore che indica se le due stringhe sono uguali
- ❑ La sintassi è:

```
int strcmp(char* str1, char* str2);
```

- ❑ La funzione restituisce un valore
 - ❖ <0 se str1 precede lessicograficamente str2
 - ❖ >0 se str1 segue lessicograficamente str2
 - ❖ =0 se str1 e str2 sono identiche

PROGRAMMA

LIBRERIA STRING.H: COMPARAZIONE

- ❑ La funzione restituisce un valore che indica se le due stringhe sono uguali
- ❑ La sintassi è:

```
int strcmp(char* str1, char* str2);
```

```
#include <stdio.h>
#include <string.h>
int main(){
    char s1[]="Paperino";
    char s2[]="Minnie";
    char s3[]="Zio Paperone";
    char s4[]="Minnie";
    int n;
    n=strcmp(s1,s2);
    printf("\n%d",n);           //Stampa 3
    n=strcmp(s2,s3);
    printf("\n%d",n);           //Stampa -13
    n=strcmp(s2,s4);
    printf("\n%d",n);           //Stampa 0
    return 0;
}
```

PROGRAMMA

LIBRERIA STRING.H: CONCATENAZIONE

- ❑ La funzione appende la stringa **src** in coda alla stringa **dest**
- ❑ La sintassi è:

```
char* strcat(char* dest, char* src);
```

```
#include <stdio.h>
#include <string.h>
int main(){
    char s1[]="Paolino ";
    char s2[]="Paperino";
    strcat(s1,s2);
    printf("\n%s",s1);
    return 0;
}
```

PROGRAMMA

LIBRERIA STRING.H: RICERCA

- ❑ La funzione ricerca la stringa sub all'interno della stringa strLa sintassi è:

```
char* strstr(char* str, char* sub);
```

- ❑ La funzione restituisce
 - ❖ L'intera stringa str se sub è la stringa vuota
 - ❖ La stringa a partire dall'inizio della prima occorrenza della sottostringa sub in str
 - ❖ NULL se la sottostringa non è presente all'interno di str

PROGRAMMA

LIBRERIA STRING.H: COMPARAZIONE

- ❑ La funzione ricerca la stringa sub all'interno della stringa str
- ❑ La sintassi è:

```
char* strstr(char* str, char* sub);
```

```
#include <stdio.h>
#include <string.h>
int main(){
    char s1[]="Paperino";
    char s2[]="ri";
    char s3[]="xx";
    char* result;
    result=strstr(s1,s2);
    printf("\n%s",result);        //rino
    result=strstr(s1,s3);
    printf("\n%s",result);        //NULL
    printf("\n%d",result);        //valore 0
    return 0;
}
```



PROGRAMMA

STRINGHE

Leggere una stringa da tastiera e eliminare le vocali

ESEMPIO

Input : "Fioriera"

Output: "Frr"



Fine