
Array di stringhe

Prof. Francesco Accarino
IIS Altiero Spinelli Sesto San Giovanni

Vettori di stringhe

- Un vettore di nomi si realizza mediante una **matrice** di tipo carattere dove ogni riga (vettore) contiene un nome. Ad esempio, per memorizzare i nomi dei giorni della settimana, si può procedere così:

```
char giorni[7][10] = {"lunedì", "martedì",  
"mercoledì", "giovedì", "venerdì", "sabato", "domenica"};
```

- dove la dimensione dei singoli vettori (tutti i vettori!), cioè 10, è determinata sommando 1 alla lunghezza del nome più lungo (**mercoledì**).
- Pertanto un vettore di stringhe è in realtà una matrice di caratteri!

Vettori di stringhe

l	u	n	e	d	ì	\0				giorni[0]
m	a	r	t	e	d	ì	\0			giorni[1]
m	e	r	c	o	l	e	d	ì	\0	giorni[2]
g	i	o	v	e	d	ì	\0			giorni[3]
v	e	n	e	r	d	ì	\0			giorni[4]
s	a	b	a	t	o	\0				giorni[5]
d	o	m	e	n	i	c	a	\0		giorni[6]

Anche se in memoria la struttura di memorizzazione è una matrice
In realtà essa può essere trattata come un vettore considerando gli
indici di riga in combinazione con il nome della matrice come
indirizzo di inizio della stringa alla quale ci si sta riferendo

Vettori di stringhe

Se si vuole gestire l'elenco degli studenti di una classe bisogna considerare il numero di studenti che compongono la classe, (che determina il numero di righe della matrice) e la lunghezza massima di un nome, (che determinerà il numero di colonne).

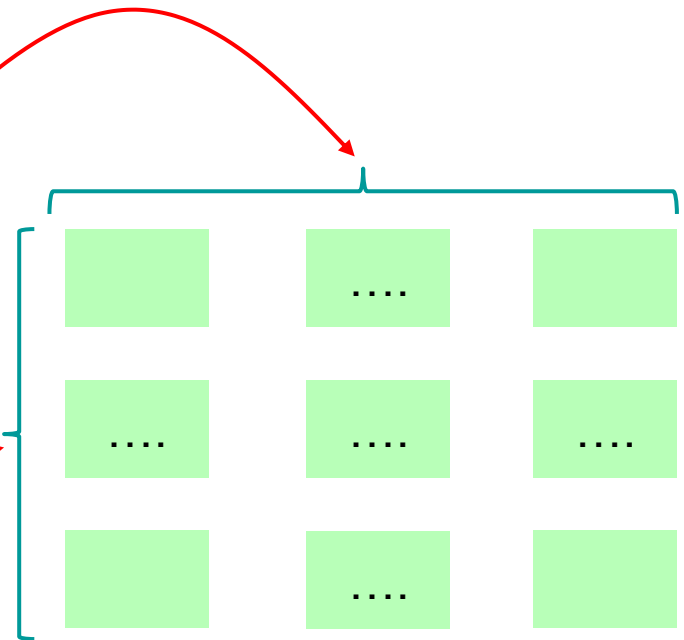
Quindi la dichiarazione della classe avverrebbe definendo innanzitutto il numero di studenti e la lunghezza massima dei nomi:

```
#define NUMSTUDENTI 20
```

```
#define MAXNOME 15
```

E poi dichiarare la classe come una matrice:

```
char classe[NUMSTUDENTI][MAXNOME];
```



Esempio Lettura dei nomi della classe

```
#include <stdio.h>
#define NUMSTUDENTI 20
#define MAXNOME 15
int main()
{
    char classe[NUMSTUDENTI][MAXNOME];
    int i;
    for(i=0;i<NUMSTUDENTI;i++)
    {
        printf("inserisci il nome dello studente: %d",i+1);
        scanf("%s",classe[i]);
    }
    return 0;
}
```

classe è dichiarata come una matrice

Ma è utilizzata come un vettore

classe[i] rappresenta l'indirizzo della i-esima riga e quindi l'indirizzo dell' i-simo nome quindi non va messa la & perché è già un indirizzo

Esempio Stampa dei nomi della classe

```
#include <stdio.h>
#define NUMSTUDENTI 20
#define MAXNOME 15
int main()
{
    char classe[NUMSTUDENTI][MAXNOME];
    int i;
    for(i=0;i<NUMSTUDENTI;i++)
    {
        printf("nome dello studente: %d %s",i+1, classe[i]);
    }
    return 0;
}
```

classe è dichiarata come una matrice

Ma è utilizzata come un vettore

classe[i] rappresenta l'indirizzo della i-esima riga e quindi dell'indirizzo dell' i-sima stringa

Esempio Ricerca lineare di un nome

```
#include <stdio.h>
#include <string.h>
#define NUMSTUDENTI 20
#define MAXNOME 15
int main()
{
    char classe[NUMSTUDENTI][MAXNOME];
    char nome[MAXNOME];
    bool trovato = false;
    int i;
    printf("inserisci il nome da cercare");
    scanf("%s",nome);
    for(i=0;i<NUMSTUDENTI;i++)
    {
        If(strcmp(classe[i],nome)==0)
            trovato = true;
        break;
    }

    if (trovato== true)
        printf("il nome %s : e' presente nella classe" ,nome");
    else
        printf("il nome %s : non e' presente nella classe" ,nome");
    return 0;
}
```

La struttura del codice è identica alla ricerca di un elemento in un vettore anche se come abbiamo detto precedentemente trattiamo la matrice di caratteri come un vettore di stringhe

Il confronto tra le stringhe non può essere fatto con l'operatore == ma attraverso la funzione strcmp che restituisce:

- 0 se le due stringhe sono uguali
- 1 se la prima è maggiore della seconda
- 1 se la prima è più piccola della seconda

Esempio Ricerca del minimo

```
#include <stdio.h>
#include <string.h>
#define NUMSTUDENTI 20
#define MAXNOME 15
int main()
{
    char classe[NUMSTUDENTI][MAXNOME];
    char min[MAXNOME];
    int i;
    strcpy(min,classe[0]);

    for(i=1;i<NUMSTUDENTI;i++)
    {
        If(strcmp(classe[i],min)<0)
            strcpy(min,classe[i]);
    }

    printf("il primo nome in ordine alfabetico della classe e': %s" ,min");

    return 0;
}
```

anche per la ricerca del minimo la struttura del codice è simile alla ricerca del minimo in un vettore

si assume che il primo nome sia il primo in ordine alfabetico (minimo)

se l' i-esimo nome è più piccolo di min si assegna a min l' i-esimo nome

Vettori di Stringhe

Anche per tutti gli altri algoritmi che abbiamo visto sui vettori:

- ❑ Ricerca del massimo
- ❑ Ricerca completa
- ❑ Ricerca Dicotomica

Valgono le stesse considerazioni fatte per gli esempi precedenti, ricordando che al posto degli operatori di confronto: `==, !=, <, >`, ecc

bisogna utilizzare le funzioni di manipolazione delle stringhe viste nelle lezioni precedenti.

Lascio a voi il compito di provare gli algoritmi sopra elencati