

Titolo unità didattica: Ordinamento di array, parte I

[11]

Titolo modulo : Function in C per l'ordinamento

[03-C]

Sviluppo di function in C per l'ordinamento ed esempi di utilizzo

Argomenti trattati:

- ✓ function in C per l'ordinamento per inserimento
- ✓ function in C per l'ordinamento per selezione di minimo
- ✓ function in C per l'ordinamento per selezione di massimo
- ✓ function in C per l'ordinamento di array di stringhe

Prerequisiti richiesti: AP-07-10-C, AP-09-04-C, AP-11-01-T, AP-11-02-T

```
#include <stdio.h>
void ord_inser(char[],int);
void visualizza_aC(char [],int);
void main()
{
    int n_a;
    char a[]={'p','z','a','r','b','c','m','s',
             'd','n','o','e','g','f','u','w','g','h'};
    int i;
    n_a = 18;
    printf("array non ordinato \n");
    visualizza_aC(a,n);
    ord_inser(a,n_a);
    printf("array ordinato \n");
    visualizza_aC(a,n);
}
```

```
/* function di ordinamento per inserimento
   di un array di size n */
void ord_inser(char array[], int n)
{
    int i, j;
    char el_da_ins;
    for (i=1; i<n; i++)
    {
        el_da_ins = array[i];
        j = i-1;
        while(j >= 0 && el_da_ins < array[j])
        {
            array[j+1] = array[j];
            j--;
        }
        array[j+1] = el_da_ins;
    }
}
```

```

#include <stdio.h>
void ord_sel_min(char [],int);
void min_val_ind(char [],int ,char *, int *);
void scambiare_c(char *,char *);
void visualizza_aC(char [],int);
void main()
{
    int n_a, i;
    char a[]={'p','z','a','r','b','c','m',
             's','d','n','o','e','g','f','u','w','t','h'};
    n_a = 18;
    printf("array non ordinato \n");
    visualizza_aC(a,n);
    ord_sel_min(a,n_a);
    printf("array ordinato \n");
    visualizza_aC(a,n);
}

void scambiare_c(char *c1,char *c2)
{
    char temp;
    temp = *c1; *c1 = *c2; *c2 = temp;
}

```

```
/* funzione di ordinamento per selezione di minimo di un
   array di size n */
void ord_sel_min(char array[], int n)
{
    int i, indice_min;
    char min_array;
    for (i=0; i<n-1; i++)
    {
        min_val_ind(&array[i], n-i, &min_array, &indice_min);
        scambiare_c(&array[i], &array[indice_min+i]);
    }
}

void min_val_ind(char a[], int n, char *min_array, int *i_min)
{
    int i;
    *min_array = a[0];
    *i_min = 0;
    for (i=1; i<n; i++)
        if (a[i] < *min_array)
        {
            *min_array = a[i];
            *i_min = i;
        }
}
```

```

#include <stdio.h>
void ord_sel_max(char [],int);
void max_val_ind(char [],int ,char *, int *);
void scambiare_c(char *,char *);
void visualizza_aC(char [],int);
void main()
{
    int n_a, i;
    char a[]={'p','z','a','r','b','c','m',
             's','d','n','o','e','g','f','u','w','t','h'};
    n_a = 18;
    printf("array non ordinato \n");
    visualizza_aC(a,n);
    ord_sel_max(a,n_a);
    printf("array ordinato \n");
    visualizza_aC(a,n);
}

void scambiare_c(char *c1,char *c2)
{
    char temp;
    temp = *c1; *c1 = *c2; *c2 = temp;
}

```

```

/* function di ordinamento per selezione di massimo di un
   array di size n */
void ord_sel_max(char array[],int n)
{
    int i,indice_max;
    char max_array;
    for (i=n-1;i>0;i--)
    {
        max_val_ind(&array[0],i+1,&max_array,&indice_max);
        scambiare_c(&array[i],&array[indice_max]);
    }
}

void max_val_ind(char a[],int n,char *max_array, int *i_max)
{
    int i;
    *max_array = a[0];
    *i_max = 0;
    for (i=1;i<n;i++)
        if (a[i] > *max_array)
        {
            *max_array = a[i];
            *i_max = i;
        }
}

```

```

#include <stdio.h>
void ord_sel_max(char [],int);
int max_ind(char [],int);
void scambiare_c(char *,char *);
void visualizza_aC(char [],int);
void main()
{
    int n_a, i;
    char a[]={'p','z','a','r','b','c','m',
             's','d','n','o','e','g','f','u','w','t','h'};
    n_a = 18;
    printf("array non ordinato \n");
    visualizza_aC(a,n);
    ord_sel_max(a,n_a);
    printf("array ordinato \n");
    visualizza_aC(a,n);
}

void scambiare_c(char *c1,char *c2)
{
    char temp;
    temp = *c1; *c1 = *c2; *c2 = temp;
}

```

versione 2


```
/* function di ordinamento per selezione di massimo di un
   array di size n */
void ord_sel_max(char array[],int n)
{
    int i;
    for (i=n-1;i>0;i--)
        scambiare_c(&array[i],&array[max_ind(&array[0],i+1)]);
}
```

```
int max_ind(char a[],int n)
{
    int i,i_max;
    i_max = 0;
    for (i=1;i<n;i++)
        if (a[i] > a[i_max])
            i_max = i;
    return i_max;
}
```

versione 2

ordinamento di un array di stringhe

array 2D di **char**

P	i	p	p	o							
G	a	s	t	o	n	e					
Q	u	i									
P	l	u	t	o							
z	i	o		P	a	p	e	r	o	n	e
P	a	p	e	r	i	n	o				
P	a	p	e	r	i	n	a				

```

#include <stdio.h>
#include <string.h>
void ord_sel_max_S(char [][][50],int);
int max_ind_S(char [][][50],int);
void scambiare_S(char *,char *);
void main()
{
    int n, i;
    char elenco[][50]={"Pippo", "Gastone", "Qui", "Pluto",
        "zio Paperone", "Paperino", "Paperina"};
    n = 7;
    printf("elenco non ordinato \n");
    for (i=0;i<n;i++)
        printf(" %s\n",elenco[i]);
    ord_sel_max_S(elenco,n);
    printf("\nelenco ordinato \n");
    for (i=0;i<n;i++)
        printf(" %s\n",elenco[i]);
}

void scambiare_S(char *s1,char *s2)
{
    char temp[100];
    strcpy(temp,s1);  strcpy(s1,s2);  strcpy(s2,temp);
}

```

```

/* function di ordinamento per selezione di massimo di un
   array di size n di Stringhe */
void ord_sel_max_S(char array_Stringhe[][50],int n)
{
    int i;
    for (i=n-1;i>0;i--)
        scambiare_S(array_Stringhe[i],
                    array_Stringhe[max_ind_S(array_Stringhe,i+1)]);
}
int max_ind_S(char array_Stringhe[][50],int n)
{
    int i,i_max;
    i_max = 0;
    for (i=1;i<n;i++)
        if (strcmp(array_Stringhe[i],array_Stringhe[i_max])>0)
            i_max = i;
    return i_max;
}

```

elenco non ordinato

Pippo
 Gastone
 Qui
 Pluto
 zio Paperone
 Paperino
 Paperina

elenco ordinato

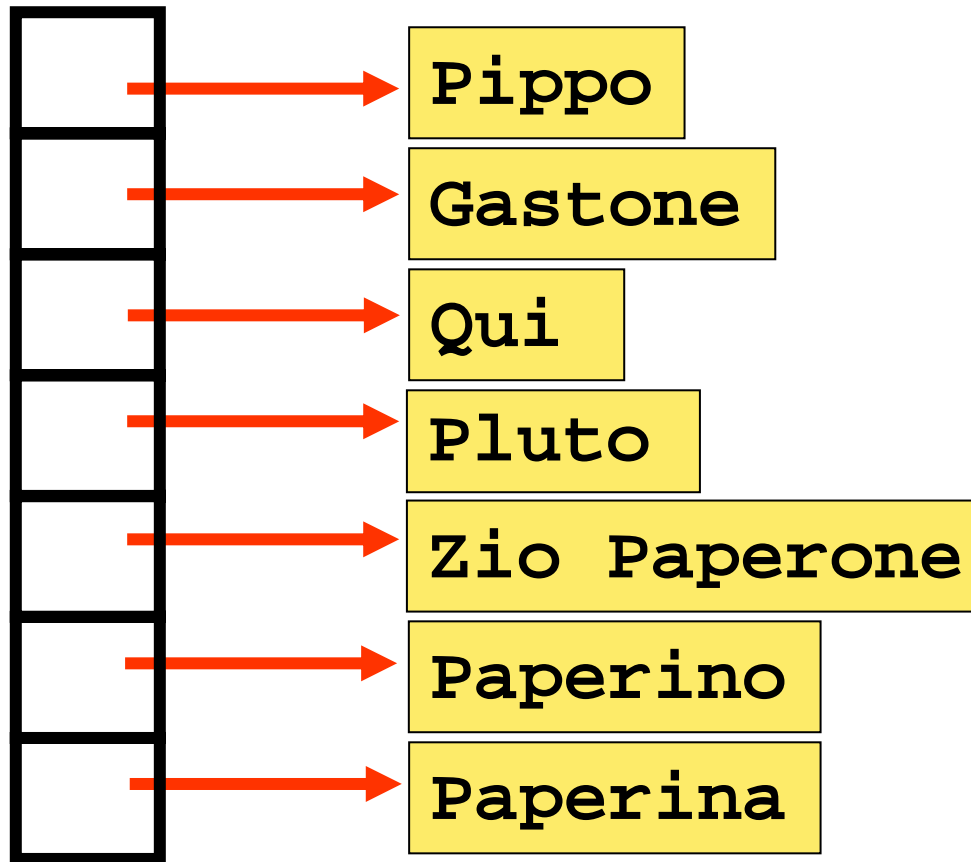
Gastone
 Paperina
 Paperino
 Pippo
 Pluto
 Qui
 zio Paperone

ordinamento di un array di stringhe

array 1D di puntatori a **char**

array di puntatori a **char**

indirizzi



costanti
stringhe

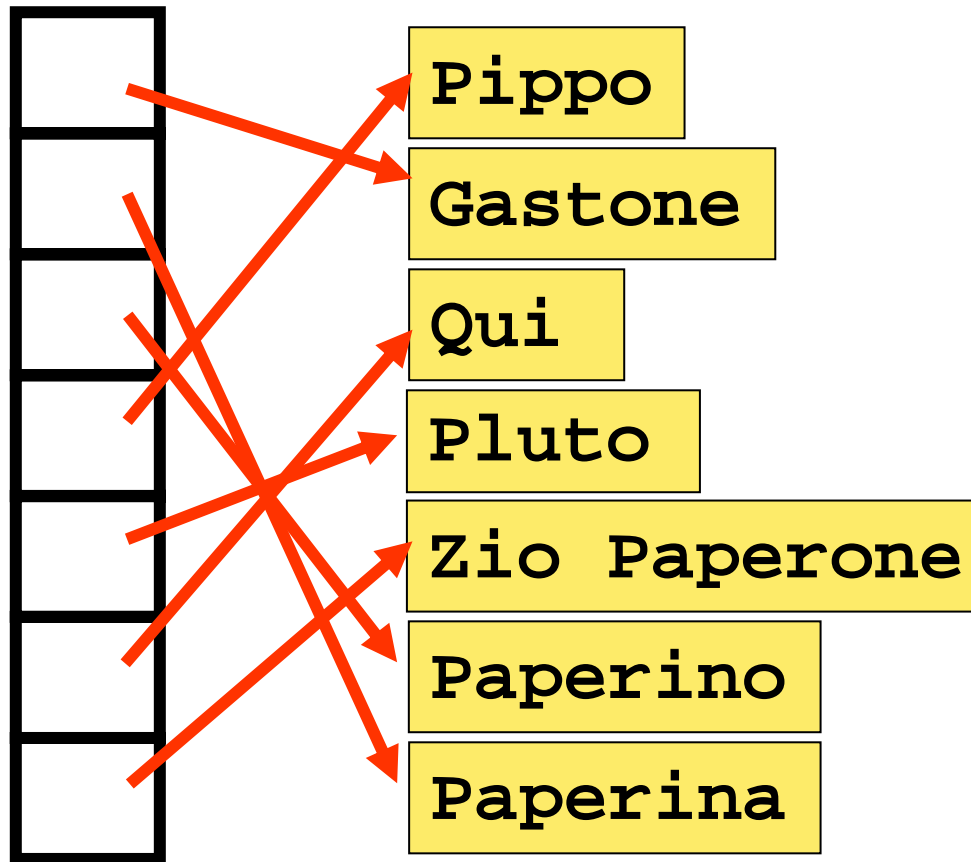
ordinamento di un array di stringhe

array 1D di puntatori a **char**

scambio di indirizzi

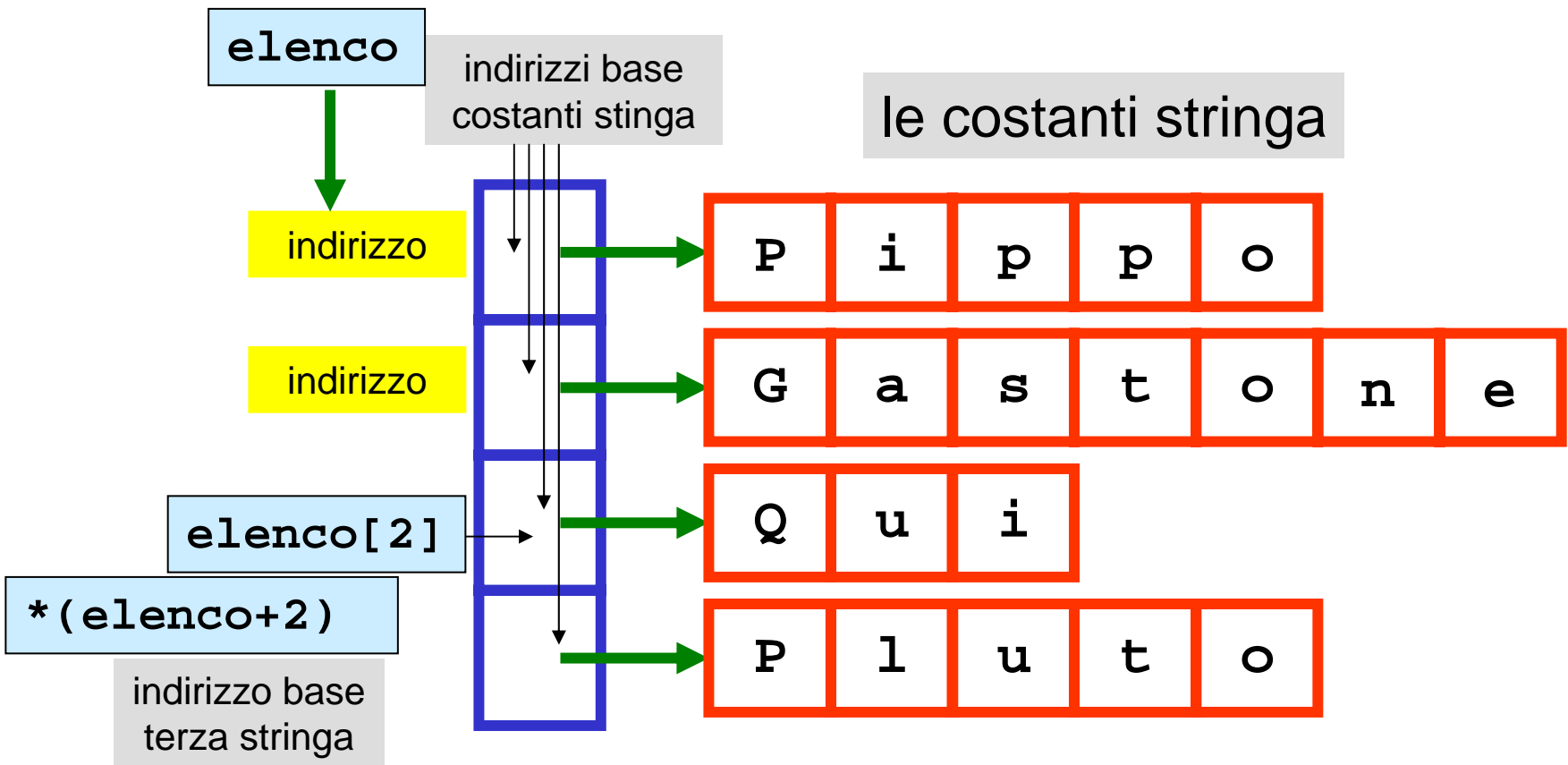
array di puntatori a **char**

indirizzi



costanti
stringhe

array frastagliato di stringhe: array di puntatori a costanti stringhe



```
#include <stdio.h>
#include <string.h>
void ord_sel_max_S(char *[],int);
int max_ind_S(char *[],int);
void scambiare_pS(char **,char **);
void main()
{
    int n, i;
    char *elenco[]={ "Pippo", "Gastone", "Qui", "Pluto",
                    "zio Paperone", "Paperino", "Paperina"};
    n = 7;
    printf("elenco non ordinato \n");
    for (i=0;i<n;i++)
        printf(" %s\n",elenco[i]);
    ord_sel_max_S(elenco,n);
    printf("\nelenco ordinato \n");
    for (i=0;i<n;i++)
        printf(" %s\n",elenco[i]);
}
```



```

/* function di ordinamento per selezione di massimo di un
   array di size n di Stringhe */
void ord_sel_max_S(char *array_Stringhe[],int n)
{
    int i;
    for (i=n-1;i>0;i--)
        scambiare_pS(&array_Stringhe[i],
                    &array_Stringhe[max_ind_S(&array_Stringhe[0],i+1)]);
}
int max_ind_S(char *array_Stringhe[],int n)
{
    int i,i_max;
    i_max = 0;
    for (i=1;i<n;i++)
        if (strcmp(array_Stringhe[i],array_Stringhe[i_max])>0)
            i_max = i;
    return i_max;
}
void scambiare_pS(char **s1,char **s2)
{
    char *temp;
    temp = *s1; *s1 = *s2; *s2 = temp;
}

```

`char **p`

puntatore a puntatore

è una variabile che contiene un **indirizzo** di un **puntatore**

`char **p`

`*p`

è un indirizzo

`**p`

è un dato

