

Function in C per problemi base con array

Titolo unità didattica: Strutture dati: array

[07]

Titolo modulo : Function in C per problemi di base con array – parte 2

[10-C]

Sviluppo di function in C per problemi di base per array 1D ed esempi di utilizzo

Argomenti trattati:

- ✓ function in C per la ricerca sequenziale
- ✓ porzioni di array in C
- ✓ utilizzo di function C su porzioni di array

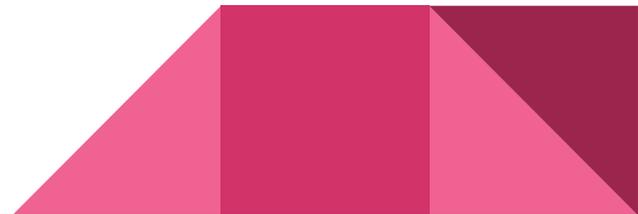
Prerequisiti richiesti: AP-05-03-C, AP-07-03-T, AP-07-08-C

Esercizio 1

Scrivere un programma in C che chiede in input n interi.
 n definito dall'utente.

Gli n interi vanno conservati in un array.

Successivamente calcola la somma e visualizza l'array



```
3 int somma(int a[], int n);
4 void visualizza(int a[], int n);
5
6
7 int main(void){
8
9     int n , i=0, sum;
10    int a[100];
11
12    printf("Inserisci il numero di elementi nell'array : \n");
13    scanf("%d", &n);
14
15    printf("\nInizia ad inserire %d elementi : \n", n);
16    for(i =0; i<n; i++){
17        scanf("%d\n", &a[i]);
18    }
19
20    sum = somma(&a[0], n);
21    printf("La somma : %d", sum);
22    visualizza(a, n);
23
24 }
25
26 void visualizza(int a[], int n){
27     int i =0;
28     for(i=0; i<n; i++){
29         printf("%d - ", a[i]);
30     }
31     printf("\n");
32 }
33
34 int somma(int a[], int n){
35     int i =0, somma = 0;
36     for(i=0; i<n; i++){
37         somma += a[i];
38     }
39     return somma;
40 }
```

```
int main(void){
    // sommare dall'elemento 12 al 17

    int n=20, i=0, sum;
    int a[100];

    //printf("Inserisci il numero di elementi nell'array : \n");
    //scanf("%d", &n);

    printf("Inizia ad inserire %d elementi : \n", n);
    for(i =0; i<n; i++){
        scanf("%d\n", &a[i]);
    }

    //sommare 5 elementi a partire da 12
    sum = somma(&a[12], 5);
    printf("La somma : %d\n", sum);
    visualizza(a, n);
}

int somma(int a[], int n){
    int i =0, somma = 0;
    for(i=0; i<n; i++){
        somma += a[i];
        printf("%d ", somma);
    }

    return somma;
}
```

```
int main(void){
    // sommare dall'elemento 12 al 17

    int n=20, i=0, sum;
    int a[100];

    //printf("Inserisci il numero di elementi nell'array : \n");
    //scanf("%d", &n);

    printf("Inizia ad inserire %d elementi : \n", n);
    for(i =0; i<n; i++){
        scanf("%d\n", &a[i]);
    }

    //sommare dall'elemento 12 al 17
    sum = somma_3argomenti(&a[0], 12, 17);

    printf("La somma : %d\n", sum);
    visualizza(a, n);
}

int somma_3argomenti(int a[], int start, int end){
    int i =0, somma = 0;
    for(i=start; i<end; i++){
        somma += a[i];
    }
    return somma;
}
```

esercizio

realizzare una function C che determina l'appartenenza di un dato a un insieme (array 1D)

algoritmo di ricerca sequenziale

```
int appartiene_a(float x, float a[], int n)
```

```
int appartiene_aF(float chiave, float a[], int n)
```

tipo logico
simulato

parametri di
input

Esercizio 2

Creare una function che determina l'appartenenza di un numero(chiave) a un array (a) di size n



pseudocodice

```
int appartiene(char chiave, char a[], int n) {
    int i;
    int esito_confronto;
    i = 0 ;
    esito_confronto = 0; // false
    do {
        if (chiave == a[i]) {
            esito_confronto = 1; // true
        }
        i = i+1 ;
    }
    while ( ! esito_confronto && i < n)
    return esito_confronto ;
}
```

```
int main(void){
    int belong;
    char key;
    char a[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'l', 'm', 'n', 'o', 'p'};
    key = 'z';
    belong = appartiene(key, a, 14);

    if(belong)
        printf("appartiene\n");
    else
        printf("non appartiene\n");
    return 0;
}
```

```
int appartiene(char chiave, char a[], int n) {
    int i;
    int esito_confronto;
    i = 0 ;
    esito_confronto = 0; // false
    do {
        if (chiave == a[i]) {
            esito_confronto = 1; // true
        }
        i = i+1 ;
    }
    while ( ! esito_confronto && i < n);
}
```

```
/* function che determina l'appartenenza
a un array (a) di size n
--algoritmo di ricerca sequenziale
*/
int appartiene_aF(float chiave,float a[],int n)
{
    int i;
    i = 0;
    while(chiave != a[i] && i < n-1)
        i = i+1;
    if (chiave == a[i])
        return 1;
    else
        return 0;
}
```

```
int appartiene(char chiave,char a[],int n) {
    int i;
    logical esito_confronto;
    i = 0 ;
    esito_confronto = false ;
    do {
        i = i+1 ;
        if (chiave == a[i]) {
            esito_confronto = true ;
        }
    }
    while (! esito_confronto && i < n)
    return esito_confronto ;
}
```

versione 1

```
/* function che determina l'appartenenza di un
numero(chiave) a un array (a) di size n
   --algoritmo di ricerca sequenziale--
*/
int appartiene_aF(float chiave, float a[], int n)
{
    int i;
    i = 0;
    for(i=0; i<n; i++)
        if (chiave == a[i])
            return 1;
    return 0;
}
```

esercizio

realizzare un main che chiama la function C di ricerca sequenziale `appartiene_aF`

```
#include <stdio.h>
int appartiene_aF(float ,float [],int );
void legge_da_tastiera_aF(float [], int );
void visualizza_aF (float [], int );
void main ()
{
    float a[100],chiave;
    int esito_ricerca,n_elem;
    printf("inserire il numero di elementi (<=100):");
    scanf("%d",&n_elem);
    legge_da_tastiera_aF(a,n_elem);
    printf("inserire la chiave di ricerca:");
    scanf("%f",&chiave);
    esito_ricerca = appartiene_aF(chiave,a,n_elem);
    if (esito_ricerca)
        printf ("%f appartiene all'array \n",chiave);
    else
        printf ("%f non appartiene all'array \n",chiave);
    printf ("l'array e' \n");
    visualizza_aF(a,n_elem);
}
```

il predicato è il valore di `esito_ricerca`

esercizio

realizzare un main che chiama la function C di ricerca sequenziale `appartiene_aF`

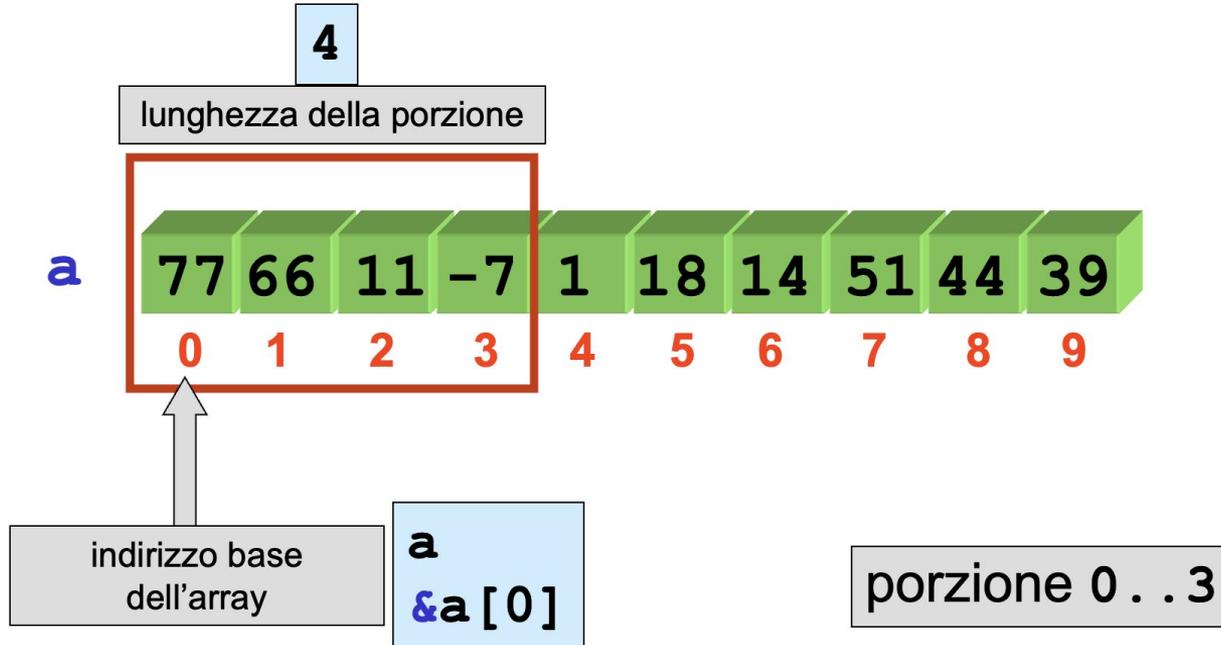
```
#include <stdio.h>
int appartiene_aF(float ,float [],int );
void legge_da_tastiera_aF(float [], int );
void visualizza_aF (float [], int );
void main ()
{
    float a[100],chiave;
    int n_elem;
    printf("inserire il numero di elementi (<=100):");
    scanf("%d",&n_elem);
    legge_da_tastiera_aF(a,n_elem);
    printf("inserire la chiave di ricerca:");
    scanf("%f",&chiave);
    if (appartiene_aF(chiave,a,n_elem))
        printf ("%f appartiene all'array \n",chiave);
    else
        printf ("%f non appartiene all'array \n",chiave);
    printf ("l'array e' \n");
    visualizza_aF(a,n_elem);
}
```

il predicato è il valore restituito dalla function



porzione di un array

modalità per specificare una **porzione** (ovvero un insieme di elementi contigui) di un array 1D, come argomento di chiamata a function)



Esercizio 3

Scrivere un main che chiama la funzione `somma(a, 4)` che somma i primi 4 elementi di un array



esercizio

realizzare un main che chiama la function C

```
int somma_arrayI(int [ ], int)
```

per calcola la somma dei primi 4 elementi di un array (porzione 0..3)

```
#include <stdio.h>
int somma_arrayI(int [ ], int);
void visualizza_aI (int [ ], int );
void main ()
{
    int a[] = {76,66,11,-7,1,18,14,51,44,39};
    int s;
    s = somma_arrayI(a,4); /* somma dei primi
                           quattro elementi di a */
    printf ("somma primi 4 elementi: %d\n",s);
    printf ("l'array e' \n");
    visualizza_aI(a,10);
}
```

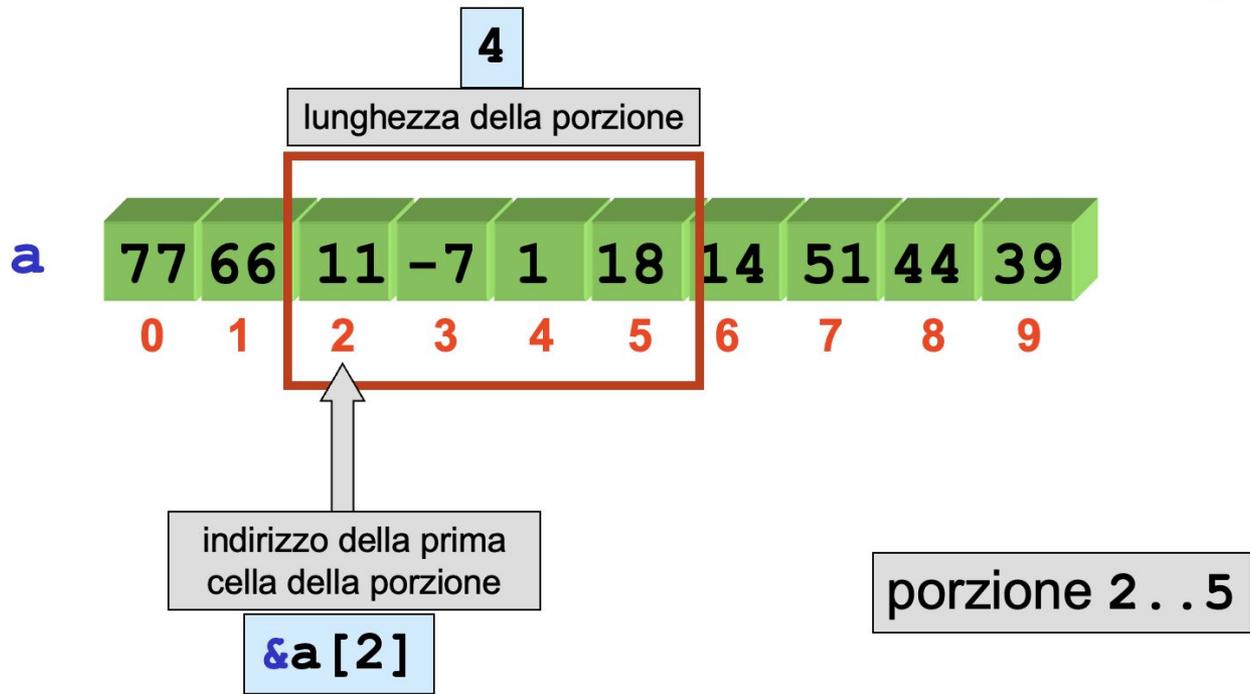
esercizio

realizzare un main che chiama la function C
`int massimo_arrayI(int [], int)`
per determinare il massimo dei primi `k` elementi
di un array (porzione `0..k-1`)

```
#include <stdio.h>
int massimo_arrayI(int [ ], int);
void legge_da_tastiera_aI(int [], int );
void visualizza_aI (int [], int );
void main ()
{
    int a[100],massimo,k,n_elem;
    printf("inserire il numero di elementi (<=100):");
    scanf("%d",&n_elem);
    legge_da_tastiera_aI(a,n_elem);
    printf("inserire k (<= %d):",n_elem);
    scanf("%d",&k);
    massimo = massimo_arrayI(a,k); /* massimo dei primi k elem. */
    printf ("il massimo tra i primi %d elementi: %d\n",k,massimo);
    printf ("l'array e'\n");
    visualizza_aI(a,n_elem);
}
```

porzione di un array

modalità per specificare una **porzione** (ovvero un insieme di elementi contigui) di un array 1D, come argomento di chiamata a function)



Esercizio 4

Scrivere un main che chiama la funzione `somma(a, 5)` che somma gli elementi della porzione 2...5 di un array



esercizio

realizzare un main che chiama la function C
`int somma_arrayI(int [], int)`
per calcola la somma degli elementi della
porzione `2..5` di un array

```
#include <stdio.h>
int somma_arrayI(int [ ], int);
void visualizza_aI (int [], int );
void main ()
{
    int a[] = {76,66,11,-7,1,18,14,51,44,39};
    int s;
    s = somma_arrayI(&a[2],4); /* somma della
    porzione di lunghezza 4 che inizia da a[2] */
    printf ("somma elementi porzione: %d\n",s);
    printf ("l'array e' \n");
    visualizza_aI(a,10);
}
```

esercizio

realizzare un main che chiama la function C

```
int massimo_arrayI(int [ ], int)
```

per determinare il massimo degli elementi della porzione

```
i_ini..i_fin
```

 di un array

```
#include <stdio.h>
int massimo_arrayI(int [ ], int);
void legge_da_tastiera_aI(int [], int );
void visualizza_aI (int [], int );
void main ()
{
    int a[100],massimo,i_ini,i_fin,n_elem;
    printf("inserire il numero di elementi (<=100):");
    scanf("%d",&n_elem);
    legge_da_tastiera_aI(a,n_elem);
    printf("inserire indice iniziale porzione(<= %d):",n_elem);
    scanf("%d",&i_ini);
    printf("inserire indice finale porzione(> %d e <= %d):",
           i_ini,n_elem);

    scanf("%d",&i_fin);
    massimo = massimo_arrayI(&a[i_ini],i_fin-i_ini+1); /* massimo
                                                         degli elementi della porzione i_ini .. i_fin */
    printf ("massimo della porzione %d .. %d: %d\n",i_ini,i_fin, massimo);
    printf ("l'array e'\n");
    visualizza_aI(a,n_elem);
}
```

esercizio

realizzare una function C che determina il massimo elemento di un array e il suo indice

```
void max_val_ind(in: float a[], int n; out: max_array i_max) {  
    int i, i_max;  
    float max_array;  
    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;  
        }  
    }  
}
```

ATTENZIONE: da modificare in C

parametri di input

```
void max_val_ind(float a[], int n,  
                float *max, int *i_max)
```

parametri di output

```
void max_val_ind(float a[],int n,  
                float *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;  
            }  
}
```

```
max_val_ind(mio_a,mio_n,&mio_max_a,&mio_i_max)
```

esercizio

realizzare un main che chiama la function C `max_val_ind` per determinare il massimo degli elementi della porzione `i_ini..i_fin` di un array e il suo indice

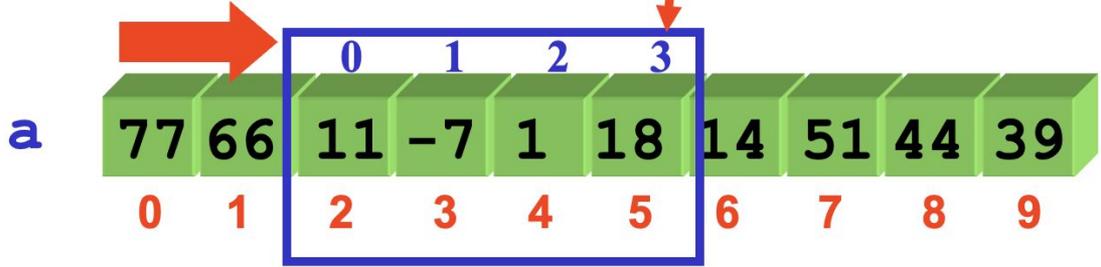
```
#include <stdio.h>
void max_val_ind(float a[],int n,float *max_array,int *i_max);
void visualizza_aF(float [], int );
void main ()
{
    float a[]={76.F,66.F,11.F,-7.F,1.F,18.F,14.F,51.f,44.f,39.F};
    float massimo;
    int i_ini,i_fin, indice_max;
    printf("inserire indice iniziale porzione(<= 10):");
    scanf("%d",&i_ini);
    printf("inserire indice finale porzione(>%d e <=10):",i_ini);
    scanf("%d",&i_fin);
    max_val_ind(&a[i_ini],i_fin-i_ini+1,&massimo,&indice_max);
    printf ("massimo della porzione %d .. %d: %f\n indice :%d",
            i_ini,i_fin,massimo,indice_max);
    printf ("l'array e'\n");
    visualizza_aF(a,10);
}
```

spiazzamento dell'indice

porzione 2..5

indice locale della porzione

```
inserire indice iniziale porzione(<= 10):2
inserire indice finale porzione(>2 e <=10):5
massimo dell' porzione 2 .. 5: 18.000000
indice :3
76.000000
66.000000
11.000000
-7.000000
1.000000
18.000000
14.000000
51.000000
44.000000
39.000000
```



il vero valore dell'indice è

`i_ini + indice_max`

indice assoluto