GLI ARRAY

Gli array

Gli array sono strutture dati statiche, di tipo sequenziale, che consentono la memorizzazione e la gestione di uno o più dati omogenei (dello stesso tipo) raggiungibili per mezzo di un indice.

Gli array

Per definire un array è necessario fornire:

- **Nome**
- **Tipo**

} **Come per le altre variabili**

- Dimensione

Dopo la dichiarazione della variabile, non può essere modificata

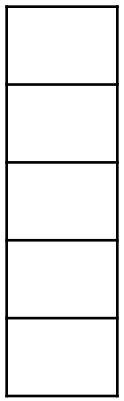
int v [5];

Gli array monodimensionali

Gli array monodimensionali o vettori sono quegli array per la cui definizione è sufficiente fornire un solo valore per la dimensione.

int v [5];

I dati vengono memorizzati in modo sequenziale, per facilitare l'accesso.



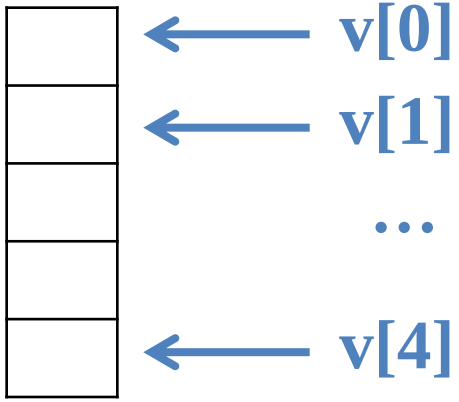
Elementi di un vettore

Per raggiungere gli elementi di un vettore è necessario precisare il nome del vettore e la posizione dell'elemento desiderato.

In C/C++ il primo elemento di un vettore si trova in posizione 0.

v[i]

Indice (numero o espressione intera positiva)



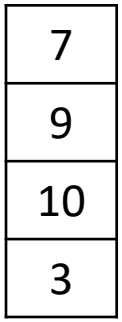
Dati singoli che possono essere usati in espressioni o assegnazioni

Assegnazione valori a vettore

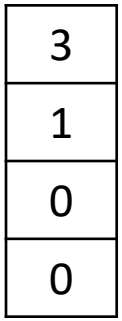
Per assegnare i valori ad un vettore è possibile:

- Assegnazione diretta

  int v[4]={7,9,10,3};

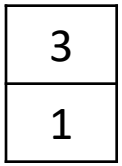


int v[4]={3,1};



~~int v[2]={3,1,9};~~

int v[]={3,1};



Meglio evitare

Assegnazione valori a vettore

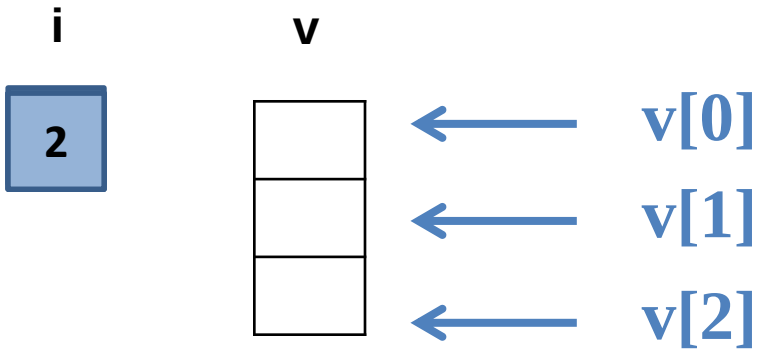
Per assegnare i valori ad un vettore è possibile:

- Assegnazione diretta
- Acquisizione da tastiera

  E' necessario assegnare i valori alle singole componenti del vettore

```
int i, v [3];

for(i=0;i<3;i++)
    scanf("%d",&v[i]);
```



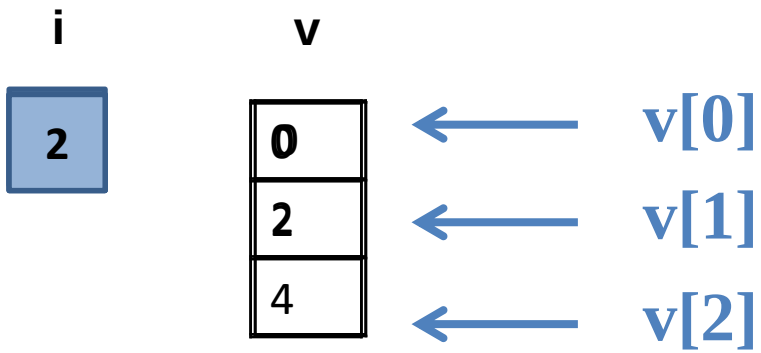
Assegnazione valori a vettore

Per assegnare i valori ad un vettore è possibile:

- Assegnazione diretta
- Acquisito da tastiera
- Successivo ad un calcolo

```
int i, v [3];

for(i=0;i<3;i++)
      v[i]=2*i;
```



Visualizzazione dei valori di un vettore

Per visualizzare i valori memorizzati in un vettore, è necessario scorrere le singole componenti e visualizzarle:

```
for(i=0;i<N;i++)
    printf(“%d”, v[i]);
```

N è la dimensione del vettore impostata in fase di dichiarazione.

Se nel programma si userà spesso N può essere utile definirla con una #define.

Copia dei valori di un vettore

Per copiare i valori di un vettore dentro un altro vettore è necessario:

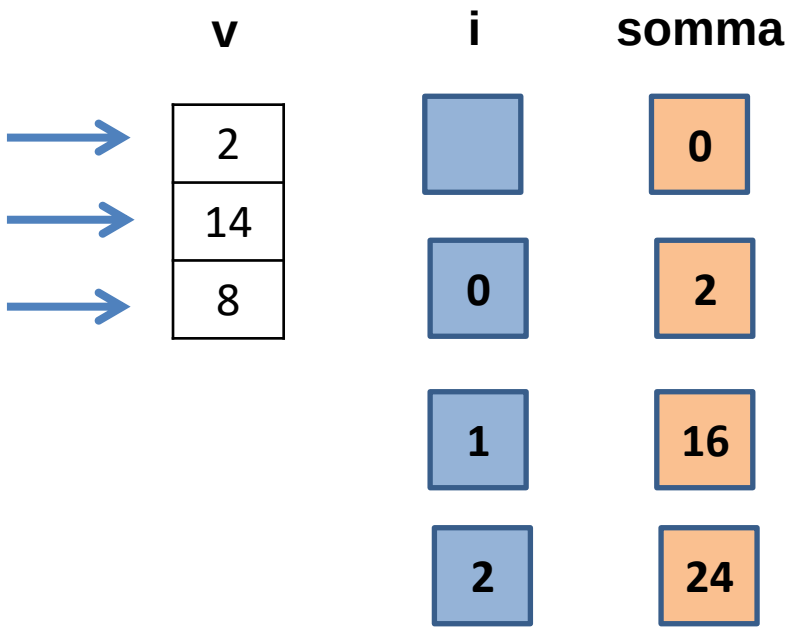
- che i due vettori abbiano la stessa dimensione oppure che si copi il vettore più piccolo in quello più grande.
- copiare le singole componenti in posizione corrispondente

```
for(i=0;i<N;i++)
    w[i]=v[i];
```

~~v=w;~~

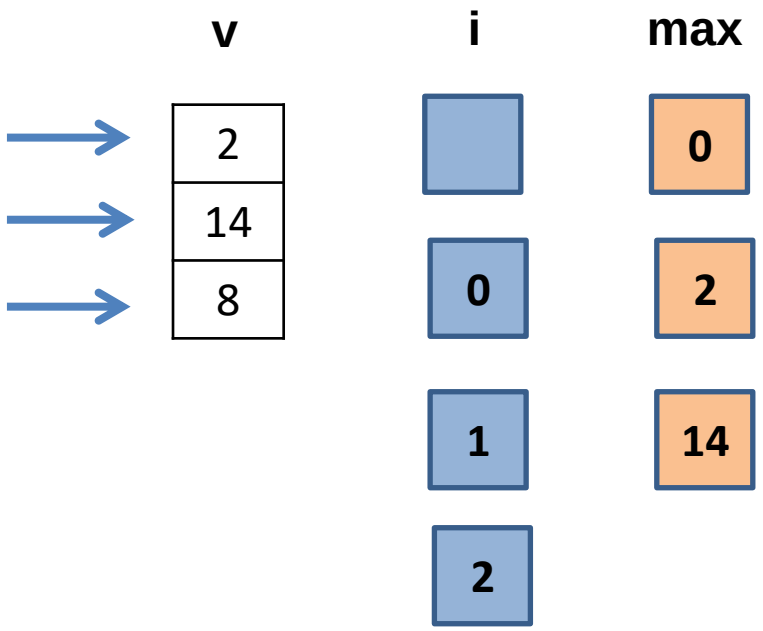
Somma degli elementi di un vettore

```
main()
{int v[3]={2,14,8};
int i, somma=0;
    for(i=0;i<3;i++)
        somma=somma+v[i];
    }
```



Ricerca dell'elemento massimo di un vettore

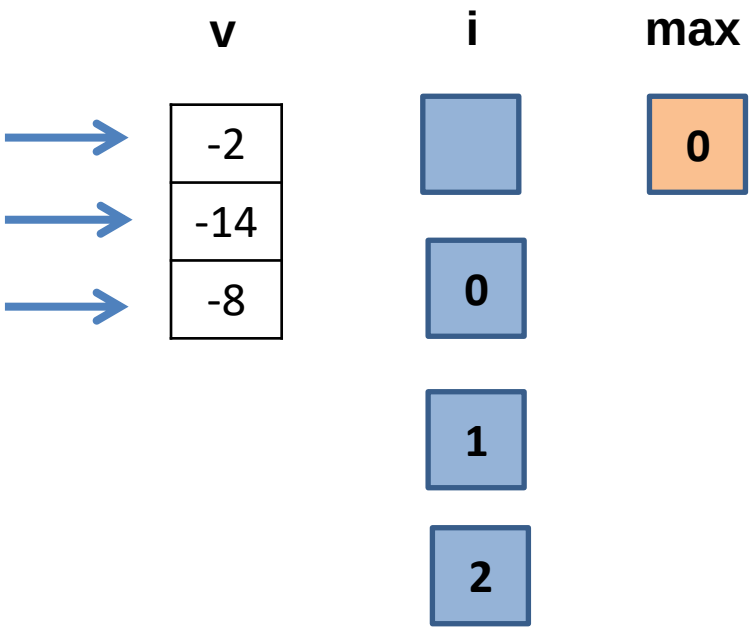
```
main()
{int v[3]={2,14,8};
int i, max=0;
    for(i=0;i<3;i++)
        if(v[i]>max)
                max=v[i];
    }
```



E se il vettore contiene numeri negativi?

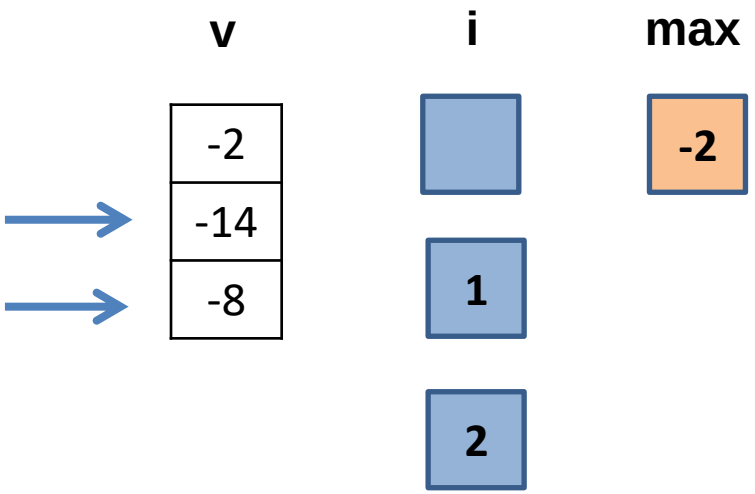
Ricerca dell'elemento massimo di un vettore

```
main()
{int v[3]={-2,-14,-8};
int i, max=0;
    for(i=0;i<3;i++)
        if(v[i]>max)
                max=v[i];
    }
```



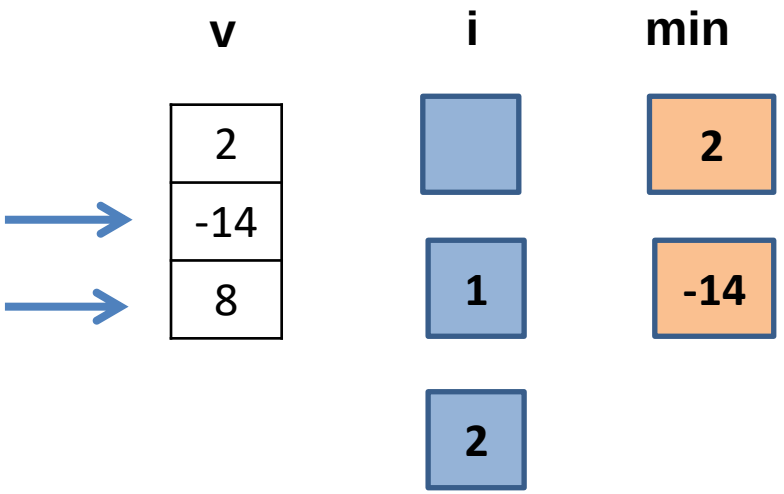
Ricerca dell'elemento massimo di un vettore

```
main()
{int v[3]={-2,-14,-8};
int i, max=v[0];
   for(i=1;i<3;i++)
        if(v[i]>max)
                max=v[i];
   }
```



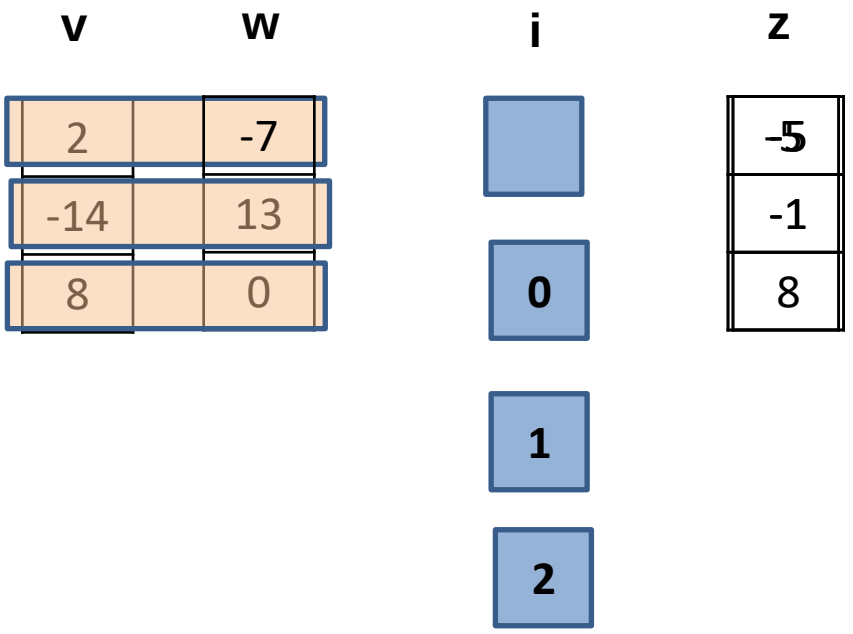
Ricerca dell'elemento minimo di un vettore

```
main()
{int v[3]={2,-14,8};
int i, min=v[0];
    for(i=1;i<3;i++)
        if(v[i]<min)
            min=v[i];
    }
```



Somma di due vettori

```
main()
{int v[3]={2,-14,8};
  int w[3]={-7,13,0};
  int i, z[3];
    for(i=0;i<3;i++)
          z[i]=v[i]+w[i];
    }
```



Questo è un esempio di uso di vettori paralleli. Si lavora su più vettori considerando gli elementi posti nella medesima posizione.

Passaggio di vettore a funzione

```
void fun(int a[10]);
main()
{int v[10];
…
fun(v);
…}
```

La dimensione può essere omessa

```
fun(int a[10])
{…}
```

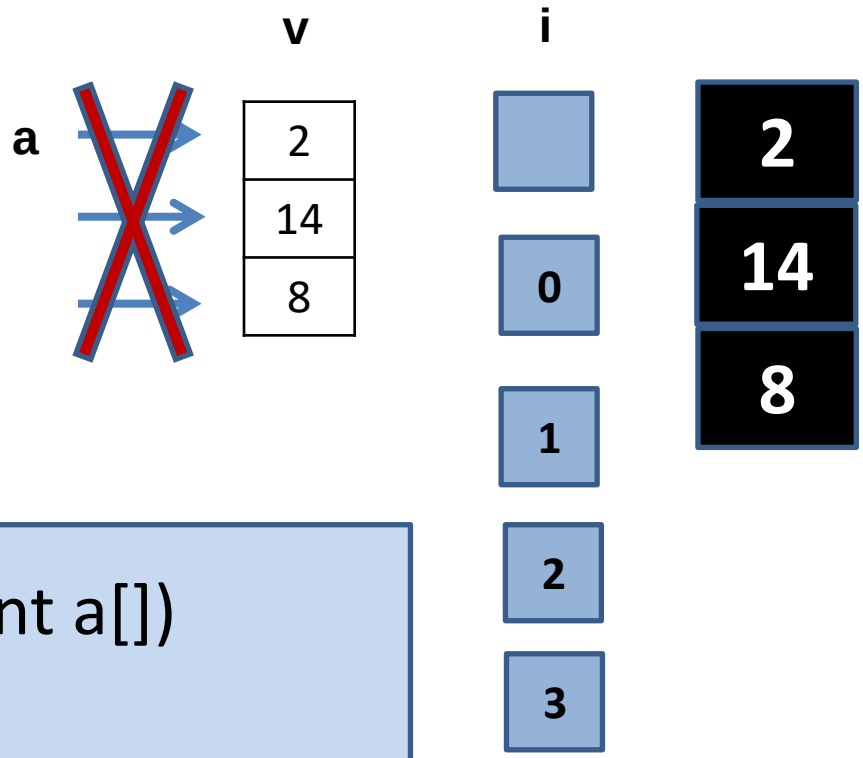
Il nome di un vettore è un puntatore alla sua prima posizione, quindi un vettore verrà sempre passato per riferimento.

Passaggio di vettore a funzione

```
#define N 3

void stampa(int a[ ]);

main()
{int v[N]={2,14,8};
stampa(v);}
```




```
void stampa(int a[])
{int i;
for(i=0;i<N;i++)
        printf(“%d\n”,a[i]);}
```

Passaggio di vettore a funzione

Il passaggio del vettore ad una funzione avviene attraverso il nome del vettore che è un puntatore alla sua prima posizione.

Il vettore viene passato per indirizzo attraverso il passaggio per valore dell'indirizzo della sua prima posizione.

E' possibile passare vettori omettendo [] nel prototipo e nella definizione ma è consigliabile metterli per facilitare la lettura del codice.

Passaggio di vettore a funzione

In C/C++ il passaggio del vettore avviene necessariamente per riferimento.

Per evitare accidentali modifiche al contenuto di un vettore è possibile ‘bloccarlo’ usando il qualificatore **const** da inserire nel prototipo e nella definizione della funzione.

```
fun(const int [])

main()
{…
fun(v);
…}
```

```
fun(const int a[])
{…}
```