



# Tipi di dato complesso

- più elementi:
  - omogenei: vettori
  - eterogenei: record, o struct

# I vettori (o array)

```
float v[10];
```

```
v[0] = 55;
```

```
v[1] = 4*v[0]-7;
```

```
scanf("%f", &v[9]);
```

```
/*scanf("%f", &v[10]);      ERRORE! */
```

```
printf("%f", &v[9]);
```

```
/* printf("%f", &v);        ERRORE! */
```

# E le stringhe?

```
char nome[256];
```

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

```
/* Attenzione: niente & e niente [] */
```

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

```
/* Attenzione: niente & e niente [] */
```

# Vettori e ciclo while

```
int valori[15];

int i;

i=0;
while (i<15) {
    printf("Inserire l'elemento %d:", i);
    scanf("%d", &valori[i]);
    i++;
}
```

Attenzione:  
(i<15) e non (i<=15)

# Vettori e ciclo while

- Questo è un caso particolare di ciclo while.

```
int valori[15];

int i;

i=0;
while (i<15) {
    printf("Inserire l'elemento %d:", i);
    scanf("%d", &valori[i]);
    i++;
}
```

# Vettori e ciclo for

- Questo è un caso particolare di ciclo while.

```
int valori[15];
int i;

i=0;
while (i<15) {
    printf("Elemento %d:", i);
    scanf("%d", &valori[i]);
    i++;
}
```

```
int valori[15];
int i;

for(i=0; i<15; i++) {
    printf("Elemento %d:", i);
    scanf("%d", &valori[i]);
}
```

Ogni ciclo for può essere tradotto in un ciclo while semanticamente equivalente!

# Da un tema d'esame

Scrivere un programma che acquisisca da tastiera 2 array contenenti 15 numeri interi (`int arr1[15]` e `int arr2[15]`) quindi generi un terzo array di interi (`risult[15]`).

L'array `risult[]` deve essere generato nel seguente modo:

- si inverta l'ordine degli elementi di `arr1[]`, generando un array `arr1_inv[]` (array in ordine inverso)
- `risult[i]` venga calcolato come somma di:  
`doppio(arr1_inv[i]) + arr2[i]`.

Il programma stampi il contenuto di `risult[]` e inoltre calcoli e stampi la media di tale vettore.



# Un altro tema d'esame

Sviluppo programma C: si acquisiscano da tastiera due array contenenti ciascuno al massimo 20 numeri interi (int a1[20], int a2[20]).

Si verifichi in fase di acquisizione che la somma degli elementi contenuti nell'array (somma1, somma2) sia positiva ( $\geq 0$ ); a tal scopo si ripeta l'acquisizione di ogni elemento N tale che la somma calcolata + N sia  $< 0$ . L'acquisizione termina non appena si inserisce un'elemento nullo o l'array è pieno (20 elementi).

Si dica quale dei 2 array ha il maggior numero di elementi e quale dei due ammette somma degli elementi minore.

NOTA: si consiglia l'utilizzo di una variabile intera di "appoggio" N in fase di acquisizione.

# Struct

- Tipo dati composito
- Formato da elementi eterogenei, ognuno identificato da un nome

```
struct persona {  
    int eta;  
    int altezza;  
    float peso;  
    char nome[256];  
}  
  
struct persona mario;  
mario.eta = 17;  
scanf("%d", &mario.altezza);
```

# Sempre più composito

- Possiamo avere anche vettori di struct...