

## Uso di funzioni

1. Introduzione e requisiti del problema
2. Progetto della soluzione
3. Codifica

## Requisiti del problema

Scrivere un programma che riceve da tastiera due successioni di DIM1 e DIM2 numeri interi distinti (in ciascuna successione) e stabilisce il valore massimo tra le 2 mediane.

Si assuma che DIM1 e DIM2 siano dispari (in questo caso la mediana si calcola come nell'esercitazione 6 "Calcolo della Mediana").

## Esempio

Date le due sequenze di lunghezza 5 e 7:

3	6	4	7	8
---	---	---	---	---

 DIM=5

2	9	3	4	8	1	7
---	---	---	---	---	---	---

 DIM=7

*la mediana della prima sequenza è 6*

*la mediana della seconda sequenza è 4*

## Esempio

Date le due sequenze di lunghezza 5 e 7:

3	6	4	7	8
---	---	---	---	---

 DIM=5

2	9	3	4	8	1	7
---	---	---	---	---	---	---

 DIM=7

*la mediana della prima sequenza è 6*

*la mediana della seconda sequenza è 4*

*il **massimo** tra le due mediane è 6*

## Funzionalità del programma

- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- Calcolo del massimo tra le mediane
- Stampa del massimo tra le mediane

## Funzionalità del programma

- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- Calcolo del massimo tra le mediane
- Stampa del massimo tra le mediane

## Funzionalità del programma

- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- **Calcolo della prima mediana**
- **Calcolo della seconda mediana**
- Calcolo del massimo tra le mediane
- Stampa del massimo tra le mediane

## Funzionalità del programma

- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- **Calcolo del massimo tra le mediane**
- Stampa del massimo tra le mediane



## Funzionalità del programma

- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- Calcolo del massimo tra le mediane
- **Stampa del massimo tra le mediane**

## Funzionalità del programma

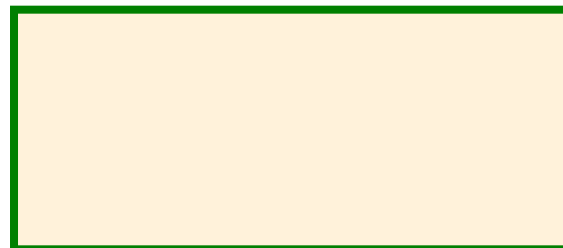
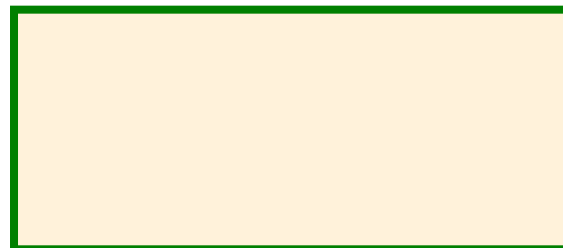
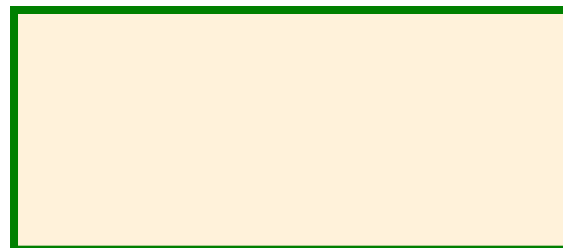
- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- Calcolo del massimo tra le mediane
- Stampa del massimo tra le mediane

## Funzionalità del programma

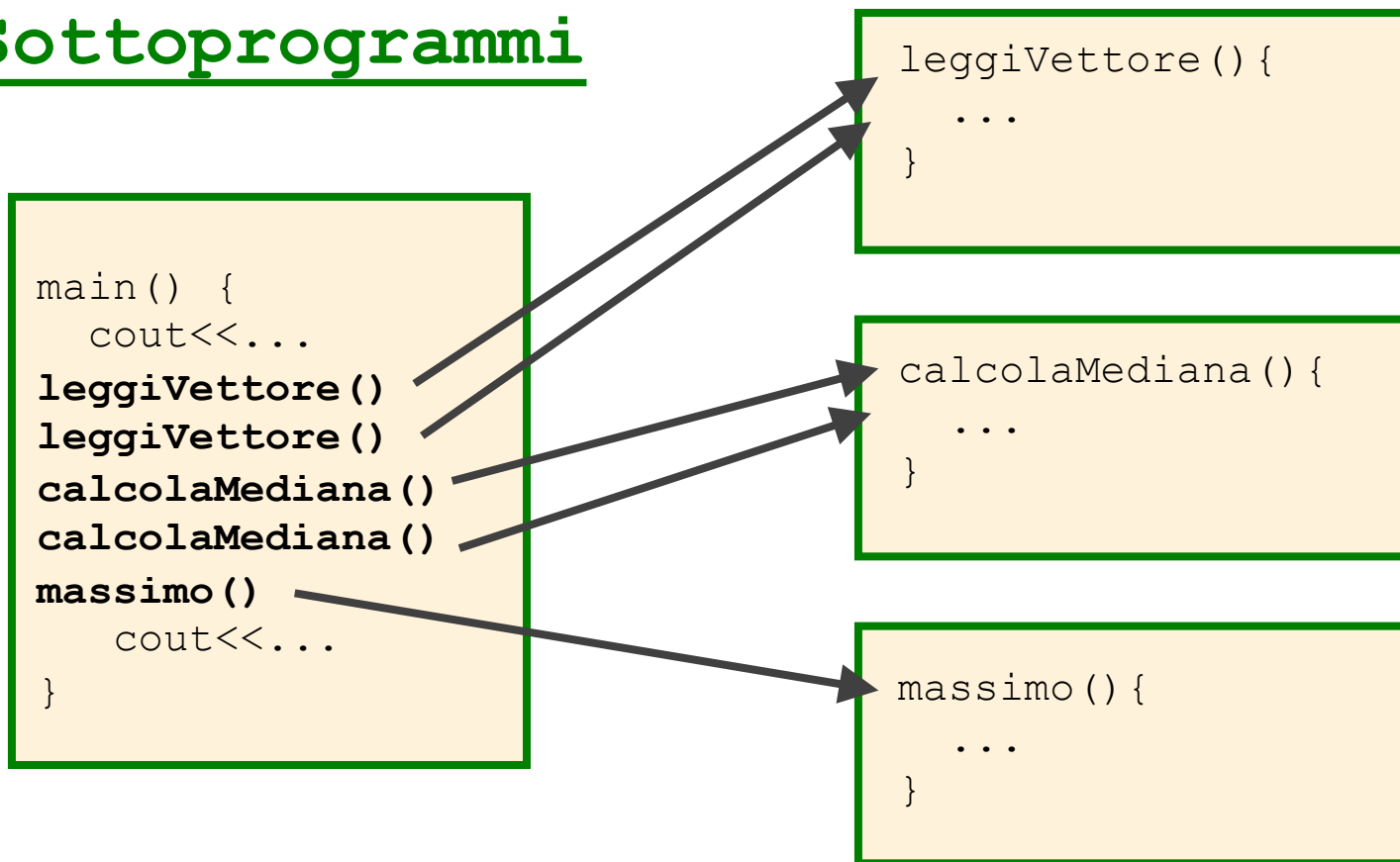
- Lettura del primo insieme di numeri
- Lettura del secondo insieme di numeri
- Calcolo della prima mediana
- Calcolo della seconda mediana
- Calcolo del massimo tra le mediane
- Stampa del massimo tra le mediane

## Sottoprogrammi

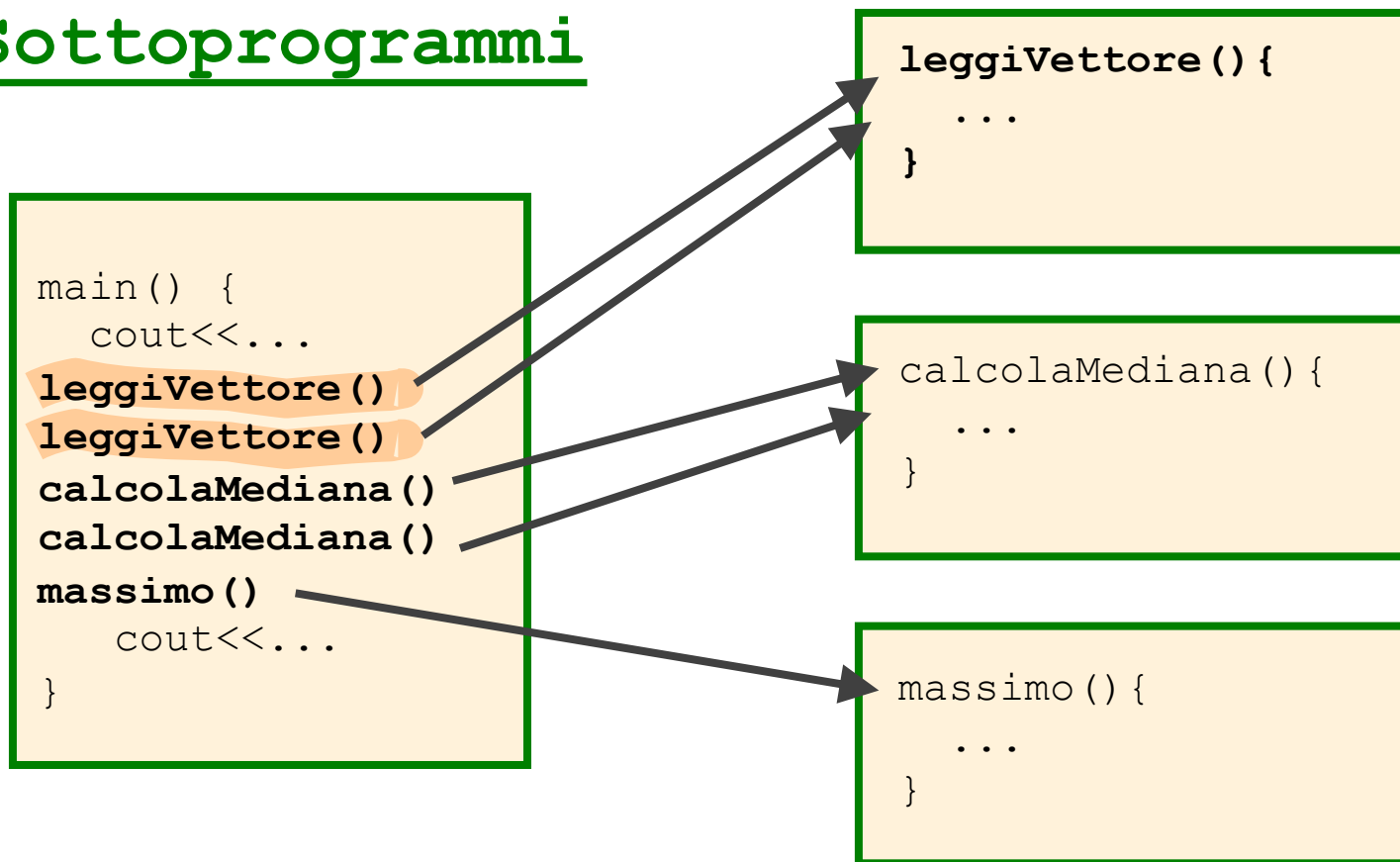
```
main() {  
    ...  
    ...  
    ...  
}
```



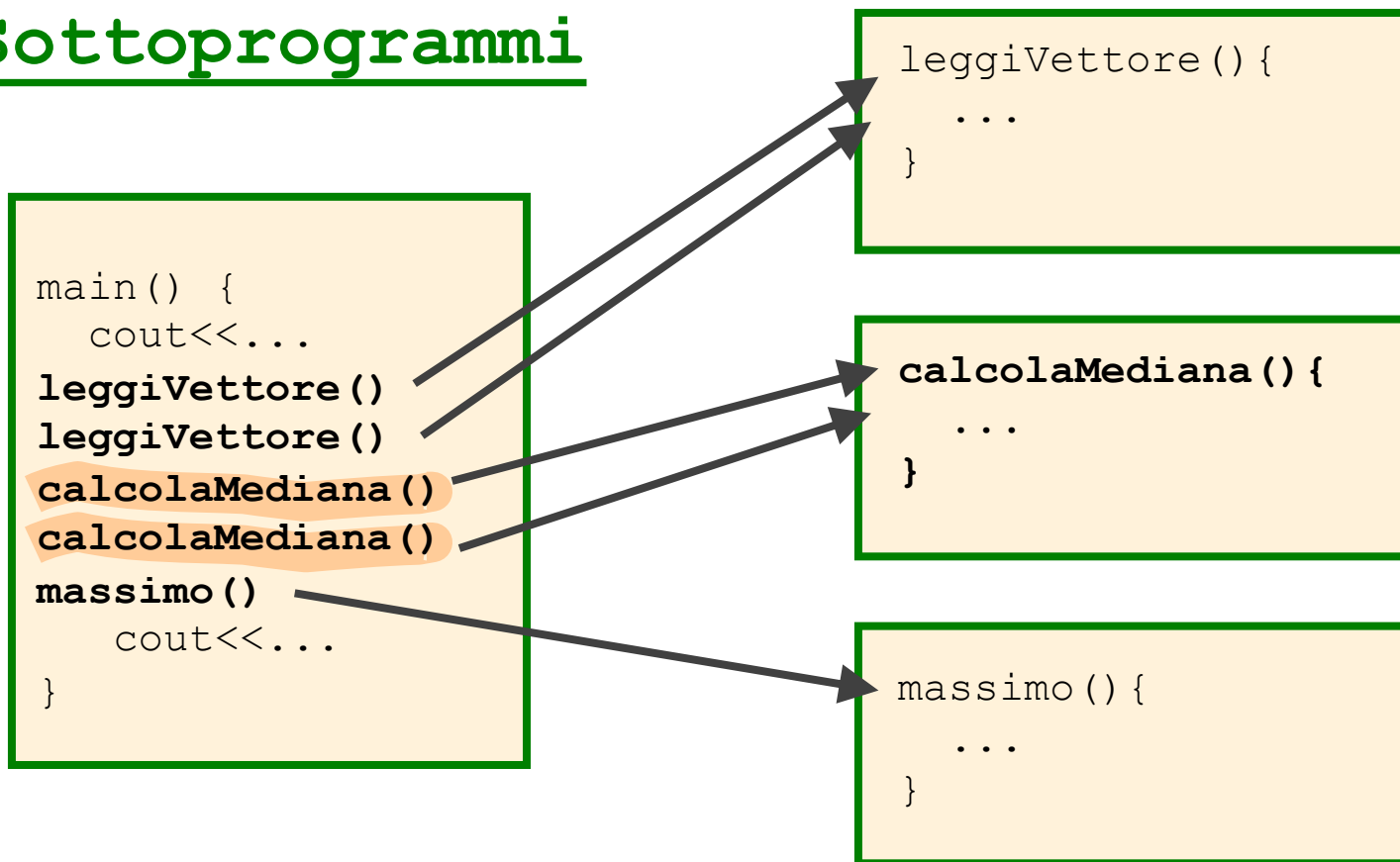
## Sottoprogrammi



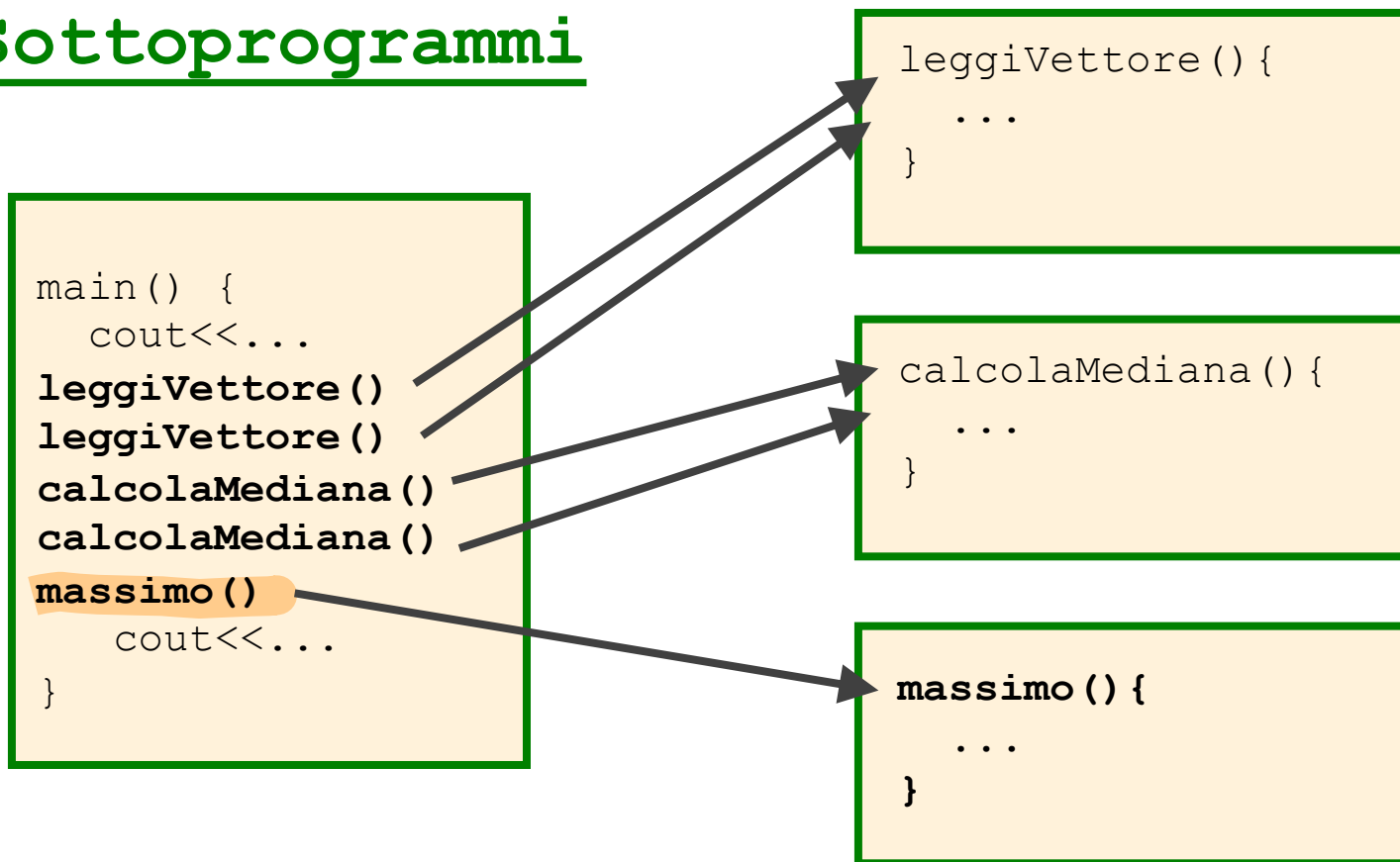
## Sottoprogrammi



## Sottoprogrammi



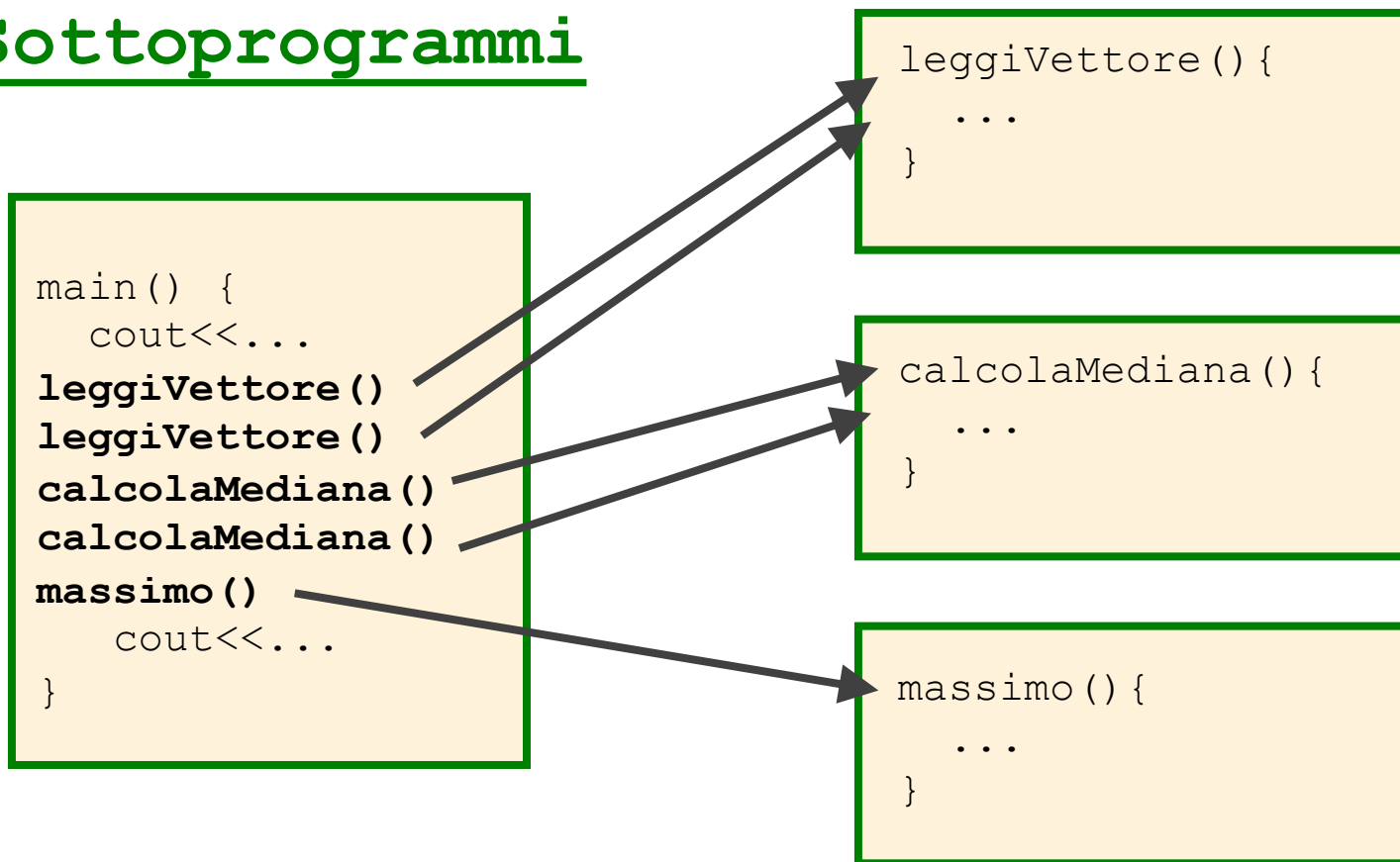
## Sottoprogrammi



*compito ben preciso*



## Sottoprogrammi



## Funzione leggiVettore

Prima sequenza: lunga DIM1 → `int insieme1[DIM1];`

Seconda sequenza: lunga DIM2 → `int insieme2[DIM2];`

```
... leggiVettore(..., ...){  
    ...  
}
```

## Funzione leggiVettore

Prima sequenza: lunga DIM1 → int insieme1[DIM1];

Seconda sequenza: lunga DIM2 → int insieme2[DIM2];

```
... leggiVettore(..., int dimensione) {  
    ...  
}
```

## Funzione leggiVettore

Prima sequenza: lunga DIM1 → `int insieme1[DIM1];`

Seconda sequenza: lunga DIM2 → `int insieme2[DIM2];`

```
... leggiVettore(..., int dimensione) {  
    ...  
}
```

## Funzione leggiVettore

Prima sequenza: lunga DIM1 → `int insieme1[DIM1];`

Seconda sequenza: lunga DIM2 → `int insieme2[DIM2];`

```
void leggiVettore(int vettore[], int dimensione){  
    ...  
}
```

**Eccezione:**

alla funzione viene passato l'indirizzo del vettore!

## Funzione leggiVettore

```
void leggiVettore(int vettore[], int dimensione){  
  
    ...  
}
```

```
main() {  
    int insieme1[DIM1],  
        insieme2[DIM2];  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana();  
    calcolaMediana();  
    massimo()  
}
```

## Funzione leggiVettore

```
void leggiVettore(int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    int insieme1[DIM1],  
        insieme2[DIM2];  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana();  
    calcolaMediana();  
    massimo();  
}
```

## Funzione leggiVettore

```
void leggiVettore(int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    int insieme1[DIM1],  
        insieme2[DIM2];  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana();  
    calcolaMediana();  
    massimo();  
}
```



## Funzione leggiVettore

```
void leggiVettore(int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    int insieme1[DIM1],  
        insieme2[DIM2];  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana();  
    calcolaMediana();  
    massimo();  
}
```

## Funzione leggiVettore

```
void leggiVettore(int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    int insieme1[DIM1],  
        insieme2[DIM2];  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana();  
    calcolaMediana();  
    massimo();  
}
```

```
insieme1=&insieme1[0]  
insieme2=&insieme2[0]
```

## Funzione calcolaMediana

```
...calcolaMediana(int vettore[], int dimensione){  
  
    ...  
}
```

```
main() {  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana(insieme1, DIM1);  
    calcolaMediana(insieme2, DIM2);  
    massimo();  
    cout<<...  
}
```

## Funzione calcolaMediana

```
...calcolaMediana(int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana(insieme1, DIM1);  
    calcolaMediana(insieme2, DIM2);  
    massimo();  
    cout<<...  
}
```

per non superare  
i limiti del vettore!

## Funzione calcolaMediana

```
...calcolaMediana(const int vettore[], int dimensione){  
    ...  
}
```

```
main() {  
    cout<<...  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    calcolaMediana(insieme1, DIM1);  
    calcolaMediana(insieme2, DIM2);  
    massimo();  
    cout<<...  
}
```

i valori del vettore non  
possono essere modificati  
dalla funzione

## Funzione calcolaMediana

### Risultato

- parametro della funzione passato per riferimento (si veda la variante 1)
- valore restituito dalla funzione (si veda la variante 2 nella codifica)

## Funzione calcolaMediana

### Risultato

- parametro della funzione passato per riferimento (si veda la variante 1)
- valore restituito dalla funzione (si veda la variante 2 nella codifica)

## Funzione calcolaMediana

### Risultato

- parametro della funzione passato per riferimento (si veda la variante 1)
- valore restituito dalla funzione (si veda la variante 2 nella codifica)



## Funzione calcolaMediana

### Risultato

- parametro della funzione passato per riferimento (si veda la variante 1)
- valore restituito dalla funzione (si veda la variante 2 nella codifica)

```
void calcolaMediana(const int vettore[], int dimensione,  
                    int & mediana){  
    ...  
}
```

non si possono costruire espressioni complesse!

## Funzione calcolaMediana

### Risultato

- parametro della funzione passato per riferimento (si veda la variante 1)
- valore restituito dalla funzione (si veda la variante 2 nella codifica)

```
int calcolaMediana(const int vettore[], int dimensione){  
    int & mediana  
    ...  
    return mediana;  
}
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int dimensione):  
  
    risultato=calcolaMediana(...)+5;
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int dimensione):  
  
    risultato=calcolaMediana(...)+5;
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int &dimensione):  
    risultato=calcolaMediana(...)+5;
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int &dimensione):  
    risultato=calcolaMediana(...)+5;
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int &dimensione):  
    risultato=calcolaMediana(...)+5;
```

## Funzione calcolaMediana

### Esempio

aggiungere 5 al valore della mediana calcolata utilizzando la funzione **calcolaMediana** nei due casi:

#### primo approccio

```
void calcolaMediana(const int vettore[], int &dimensione,  
int & mediana:
```

```
    int & mediana;  
    calcolaMediana(..., mediana);  
    risultato=mediana+5;
```

#### secondo approccio

```
int calcolaMediana(const int vettore[], int &dimensione):  
    risultato=calcolaMediana(...)+5;
```



## Funzione Massimo

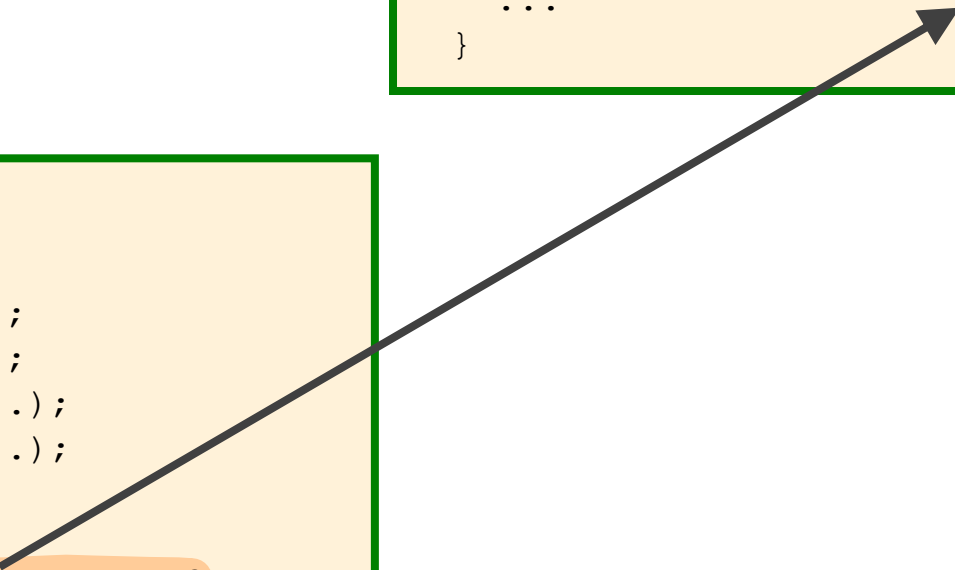
```
int massimo(int numero1, int numero2){  
    ...  
}
```

```
main() {  
    cout<<...  
    leggiVettore(...);  
    leggiVettore(...);  
    calcolaMediana(...);  
    calcolaMediana(...);  
    massimo(...);  
    cout<<  
    massimo(mediana1, mediana2);  
}
```

## Funzione Massimo

```
main() {  
    cout<<...  
    leggiVettore(...);  
    leggiVettore(...);  
    calcolaMediana(...);  
    calcolaMediana(...);  
    massimo(...);  
    cout<<  
    massimo(mediana1, mediana2);  
}
```

```
int massimo(int numero1, int numero2) {  
    ...  
}
```



## Funzione Massimo

*primo approccio*

```
int mediana1, mediana2;  
calcolaMediana(insieme1, DIM1, mediana1);  
calcolaMediana(insieme2, DIM2, mediana2);  
cout<<massimo(mediana1, mediana2) << endl;
```

## Funzione Massimo

*secondo approccio*

```
cout<<massimo(calcolaMediana(insieme1, DIM1),  
calcolaMediana(insieme2, DIM2)) << endl;
```

## Funzione Massimo

primo approccio

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    int mediana1, mediana2;  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    mediana1=calcolaMediana(insieme1, DIM1);  
    mediana2=calcolaMediana(insieme2, DIM2);  
    cout << massimo(mediana1, mediana2);  
}
```

```
void leggiVettore(int vettore[],  
                  int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

*primo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    int mediana1, mediana2;  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    mediana1=calcolaMediana(insieme1, DIM1);  
    mediana2=calcolaMediana(insieme2, DIM2);  
    cout << massimo(mediana1, mediana2);  
}
```

```
void leggiVettore(int vettore[],  
                  int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

primo approccio

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    int medianal, mediana2;  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    medianal=calcolaMediana(insieme1, DIM1);  
    mediana2=calcolaMediana(insieme2, DIM2);  
    cout << massimo(medianal, mediana2);  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

primo approccio

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    int mediana1, mediana2;  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    mediana1=calcolaMediana(insieme1, DIM1);  
    mediana2=calcolaMediana(insieme2, DIM2);  
    cout << massimo(mediana1, mediana2);  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ) {  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2) {  
    ...  
}
```



## Funzione Massimo

primo approccio

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    int mediana1, mediana2;  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    mediana1=calcolaMediana(insieme1, DIM1);  
    mediana2=calcolaMediana(insieme2, DIM2);  
    cout << massimo(mediana1, mediana2);  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

### *secondo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    cout << massimo (  
        calcolaMediana(insieme1, DIM1),  
        calcolaMediana(insieme2, DIM2));  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

### *secondo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    cout << massimo (  
        calcolaMediana(insieme1, DIM1),  
        calcolaMediana(insieme2, DIM2));  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ){  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione){  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2){  
    ...  
}
```

## Funzione Massimo

### *secondo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    cout << massimo (  
        calcolaMediana(insieme1, DIM1),  
        calcolaMediana(insieme2, DIM2));  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ) {  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione) {  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2) {  
    ...  
}
```

## Funzione Massimo

### *secondo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    cout << massimo (  
            calcolaMediana(insieme1, DIM1),  
            calcolaMediana(insieme2, DIM2));  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ) {  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione) {  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2) {  
    ...  
}
```

## Funzione Massimo

### *secondo approccio*

```
main() {  
    int insieme1[DIM1];  
    int insieme2[DIM2];  
    leggiVettore(insieme1, DIM1);  
    leggiVettore(insieme2, DIM2);  
    cout << massimo (  
        calcolaMediana(insieme1, DIM1),  
        calcolaMediana(insieme2, DIM2));  
}
```

```
void leggiVettore(int vettore[],  
                 int dimensione ) {  
    ...  
}
```

```
int calcolaMediana(const int  
vettore[], int dimensione) {  
    ...  
}
```

```
int massimo(int numero1,  
            int numero2) {  
    ...  
}
```