

Fondamenti di Programmazione (A)

I6 - Funzioni (passaggio di parametri)

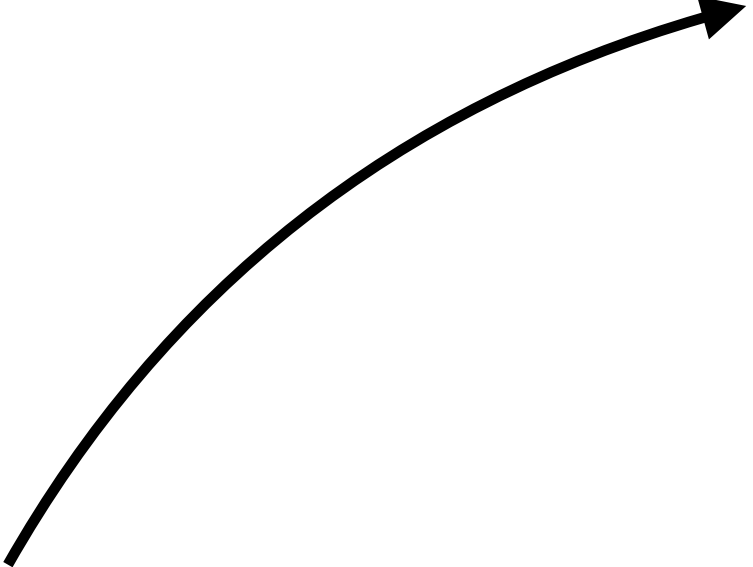
Vincenzo Arceri - Università degli Studi di Parma - vincenzo.arceri@unipr.it

Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Puntate precedenti

Signature di una funzione

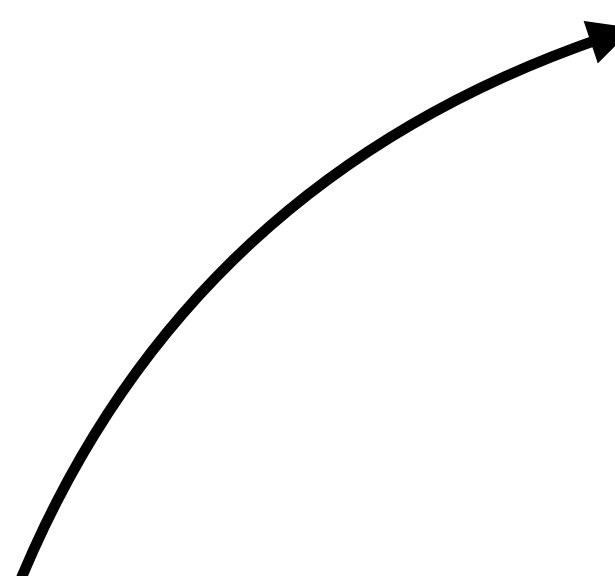


```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```

```
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Puntate precedenti

Tipo di ritorno di una funzione



```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```


Parametri formali di una funzione



```
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```



Corpo di una funzione

```
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```

```
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Chiamata di funzione

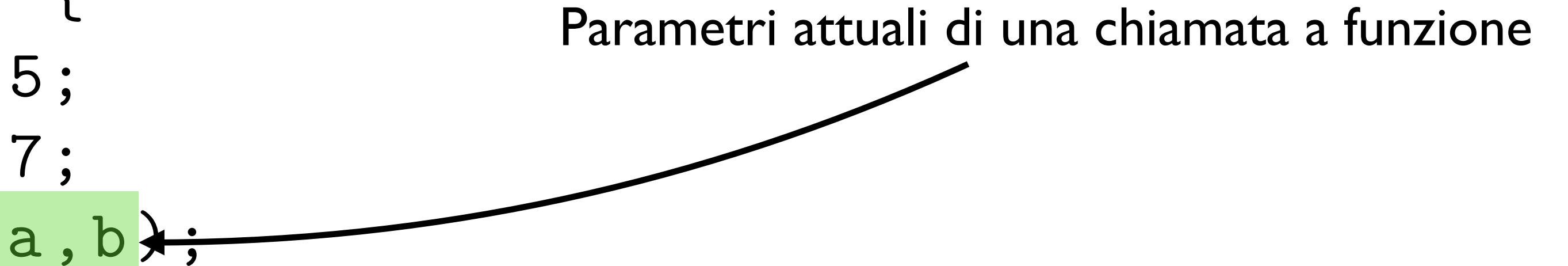


Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}
```

```
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a, b);  
    cout << a << " " << b << endl;  
}
```

Parametri attuali di una chiamata a funzione



Puntate precedenti

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Output?

Passaggio di parametri

Passaggio per valore

- Passaggio d'informazione fra *parametri attuali* e *parametri formali*
- **Passaggio per valore:** i valori dei parametri attuali vengono copiati nelle variabili corrispondenti che costituiscono i parametri formali
- In C++, le variabili vengono passate **per valore:** modificare il valore di un parametro formale **non** modifica il valore del corrispondente parametro attuale

Esempio

```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Passaggio di parametri

Passaggio per riferimento

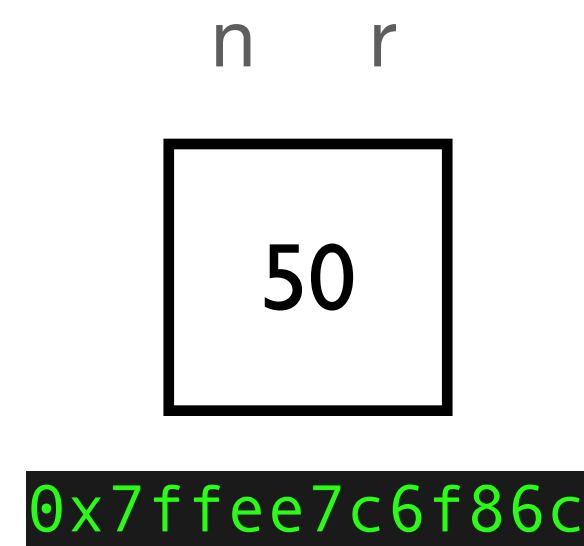
```
void scambia(int x, int y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

- Non voglio passare i valori di a e b ma voglio passare **la variabile stessa**

Tipo riferimento

```
int n = 50;  
int& r = n;
```

- La variabile `r` è un **riferimento** alla variabile `n`
- `r` è un modo alternativo per riferirsi alla variabile `n` (**alias**)



Tipo riferimento

Esempio

```
int n = 50;  
int& r = n;
```

```
n++;  
cout << n << endl;  
cout << r << endl;
```

```
int n = 50;  
int& r = n;
```

```
r++;  
cout << n << endl;  
cout << r << endl;
```

Esempio

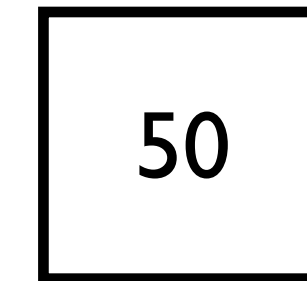
```
int  n  =  50;  
int& r1  =  n;  
int& r2  =  r1;  
  
n++;  
r1++;  
r2++;  
cout << n << endl;  
cout << r1 << endl;  
cout << r2 << endl;
```

Esempio

```
int n = 50;  
int& r1 = n;  
int& r2 = r1;
```

```
n++;  
r1++;  
r2++;  
cout << n << endl;  
cout << r1 << endl;  
cout << r2 << endl;
```

n r1 r2



0x7fee7c6f86c

Tipo riferimento

Dichiarazione

```
int n = 50;  
int& r = n;
```

- La dichiarazione di una variabile riferimento a tipo **deve avvenire contestualmente** alla sua inizializzazione

```
int n = 50;  
int& r;  
r = n;
```



Passaggio di parametri

Passaggio per riferimento

- Passaggio d'informazione fra *parametri attuali* e *parametri formali*
- **Passaggio per riferimento:** i parametri formali sono **referimenti (alias)** dei parametri attuali
- In C++, le variabili possono essere passate **per riferimento:** modificare il valore di un parametro formale **modifica** il valore del corrispondente parametro attuale

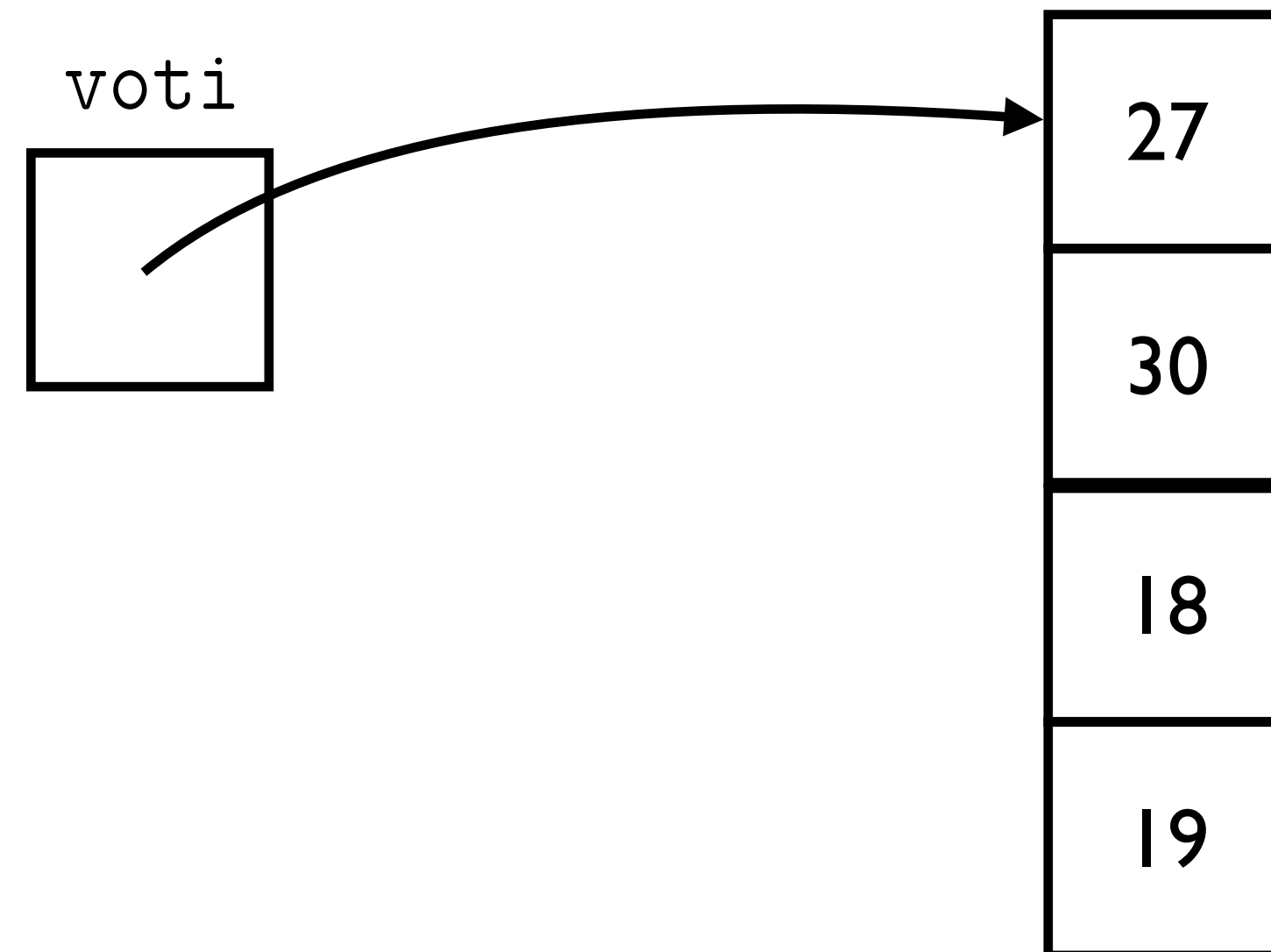
Esempio

Passaggio per riferimento

```
void scambia(int& x, int& y) {  
    int tmp = x;  
    x = y;  
    y = tmp;  
}  
  
int main() {  
    int a = 5;  
    int b = 7;  
    scambia(a,b);  
    cout << a << " " << b << endl;  
}
```

Passaggio di parametri di tipo array

```
int voti[4] = {27, 30, 18, 19}
```



Passaggio di parametri di tipo array

- **Problema:** scrivere una funzione `inverti` che, preso in input un array e due indici i e j , scambi i valori in posizione i e j dell'array

Passaggio di parametri di tipo array

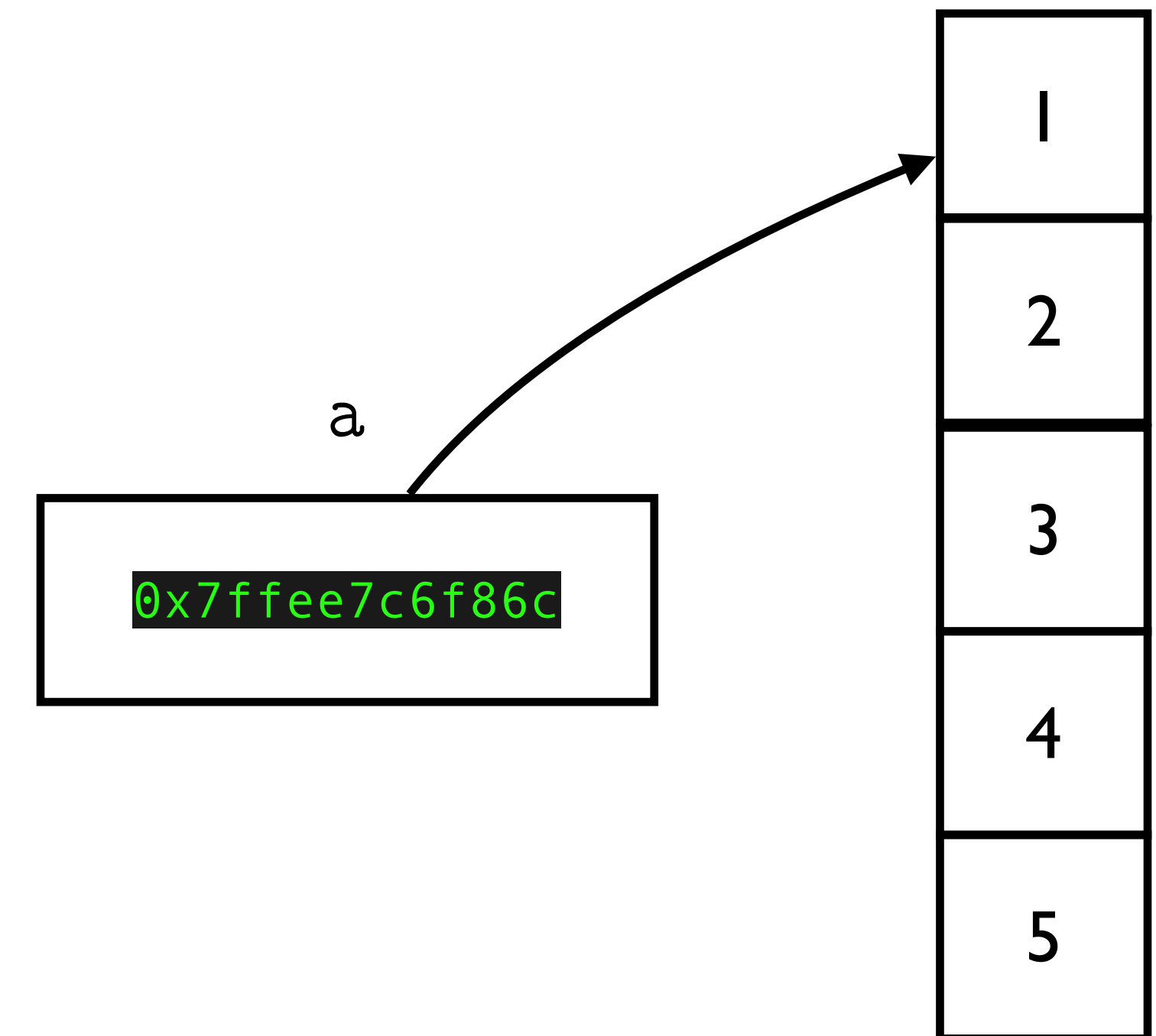
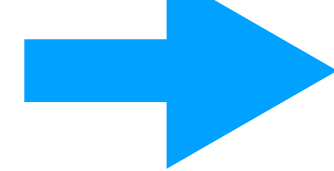
```
void inverti(int arr[], int i, int j) {  
    int tmp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = tmp;  
}  
  
int main() {  
    int a[5] = {1, 2, 3, 4, 5};  
    inverti(a, 0, 1);  
}
```

Passaggio di parametri di tipo array

- In C++ gli array vengono **passati per valore**, passando l'indirizzo del primo elemento dell'array

```
void inverti(int arr[], int i, int j) {  
    int tmp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = tmp;  
}
```

```
int main() {  
    int a[5] = {1, 2, 3, 4, 5};  
    inverti(a, 0, 1);  
}
```



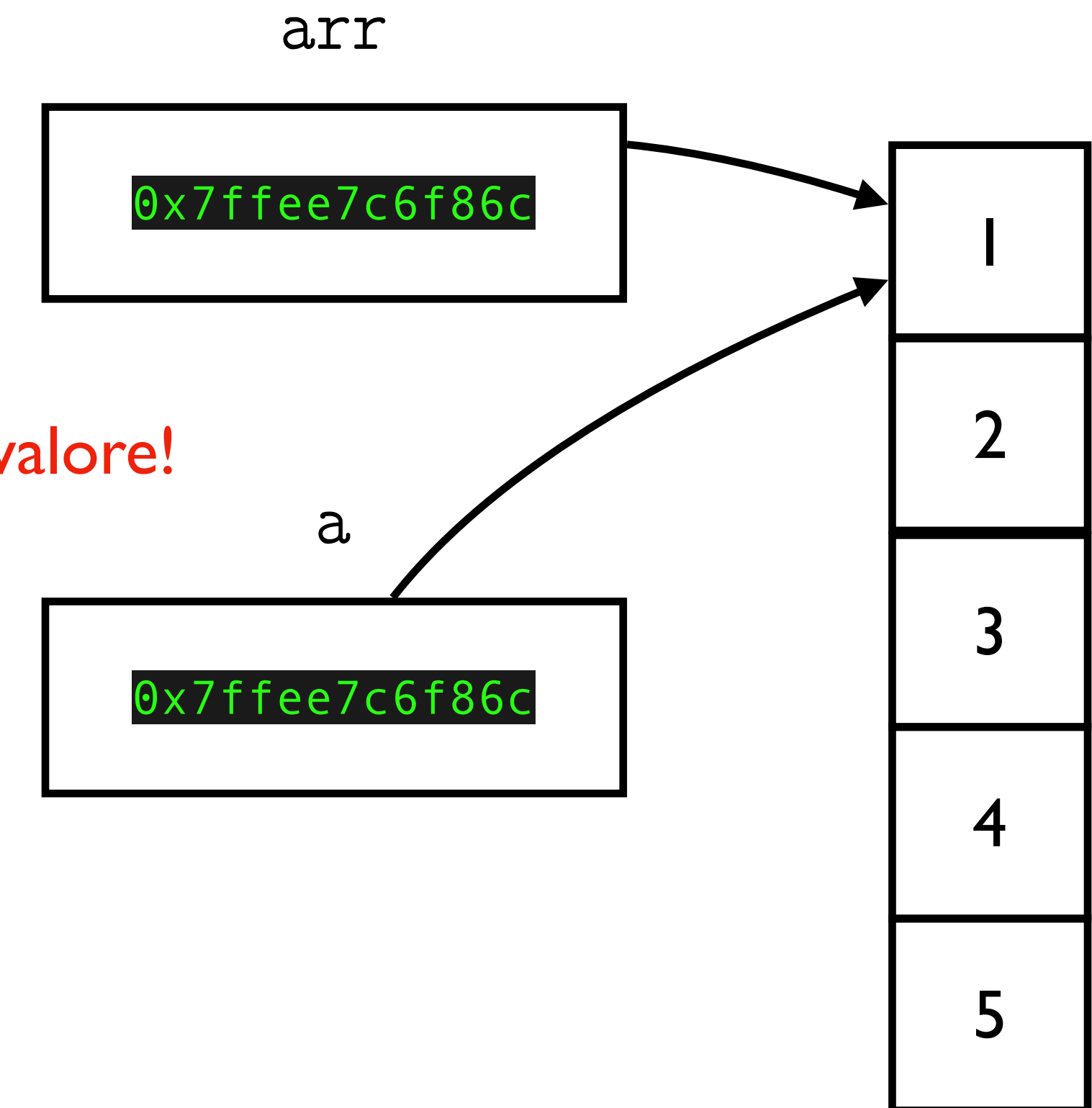
Passaggio di parametri di tipo array

- In C++ gli array vengono **passati per valore**, passando l'indirizzo del primo elemento dell'array

```
void inverti(int arr[], int i, int j) {  
    int tmp = arr[i];  
    arr[i] = arr[j];  
    arr[j] = tmp;  
}
```

```
int main() {  
    int a[5] = {1, 2, 3, 4, 5};  
    inverti(a, 0, 1);  
}
```

Passaggio per valore!



Passaggio di parametri

In C

- C passa i parametri **esclusivamente per valore**
- Non prevede il passaggio per riferimento
- Emulato tramite i puntatori
- Esempio: funzione `scambia`

Esercizio

- **Problema:** scrivere una funzione di nome `eq2` che calcola le radici di un'equazione di secondo grado

$$ax^2 + bx + c = 0$$

- Se il discriminante è negativo la funzione ritorna `false`, altrimenti `true`
- Scrivere una funzione `main` che chiami in modo opportuno la funzione `eq2`

Esercizio

- **Problema:** Scrivere una funzione di nome `contains` che presi come parametri un array di interi `A` e un intero `x`, determina se $x \in A$.

Esercizio

- **Problema:** Scrivere una funzione di nome `contains` che presi come parametri una stringa `s` e un carattere `x`, determina se $x \in s$.

Esercizio

- **Problema:** si scriva una funzione che prende in input un intero positivo n e un carattere c e stampare un triangolo rettangolo di c di lato n . Di default, il carattere c è il carattere asterisco

$n = 5, c = k$

```
k
k k
k k k
k k k k
k k k k k
```

$n = 5$

```
*
* *
* * *
* * * *
* * * * *
```