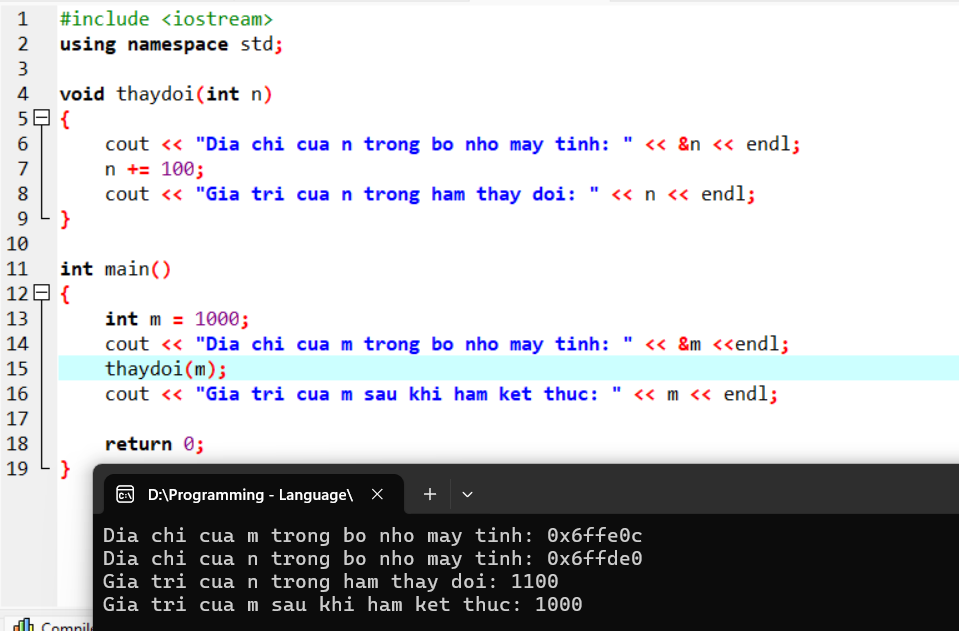
**1. Truyền Tham Trị**

**Truyền tham trị**- Pass by value là hình thức xây dựng tham số phổ biến cho hàm, như trong bài học trước về hàm thì mình đã chú ý về việc giá trị của đối số sẽ được gán cho tham số khi lời gọi hàm được thực hiện.

Xem xét ví dụ sau bạn sẽ hiểu rõ hơn về tham số và đối số, từ đó sẽ nắm được logic khi gọi hàm với cách truyền tham trị

Ở đây mình dùng toán tử địa chỉ **(&)** để in ra địa chỉ của biến trong bộ nhớ máy tính, địa chỉ thường được lưu dưới dạng số hexa.



**Giải thích :**

* n là tham số và m là đối số, giá trị của đối số m được gán cho n khi bạn gọi hàm thaydoi(m). Ở đây giá trị của m là 1000 sẽ được gán cho n và vì thế n cũng bằng 1000
* Địa chỉ của m trong máy tính là 0x6ffe0c, trong khi đó địa chỉ của n trong hàm thaydoi là 0x6ffde0 điều này chứng tỏ n và m là 2 biến khác nhau, không cùng quản lý 1 ô nhớ trong máy tính
* Trong hàm thaydoi giá trị của n được tăng lên 100, khi đó n thành 1100 và ô nhớ 0x6ffde0 (n đang quản lý) sẽ có giá trị là 1100
* Sau khi hàm thaydoi kết thúc thì giá trị của m vẫn sẽ là 1000, ô nhớ 0x6ffe0c không hề bị can thiệp gì khi hàm thay đổi thực thi

**Kết luận** : Khi hàm có tham số là tham trị thì đối số bạn truyền vào hàm và tham số của hàm là 2 biến khác nhau và việc bạn thay đổi tham số sẽ không ảnh hưởng gì tới đố số.

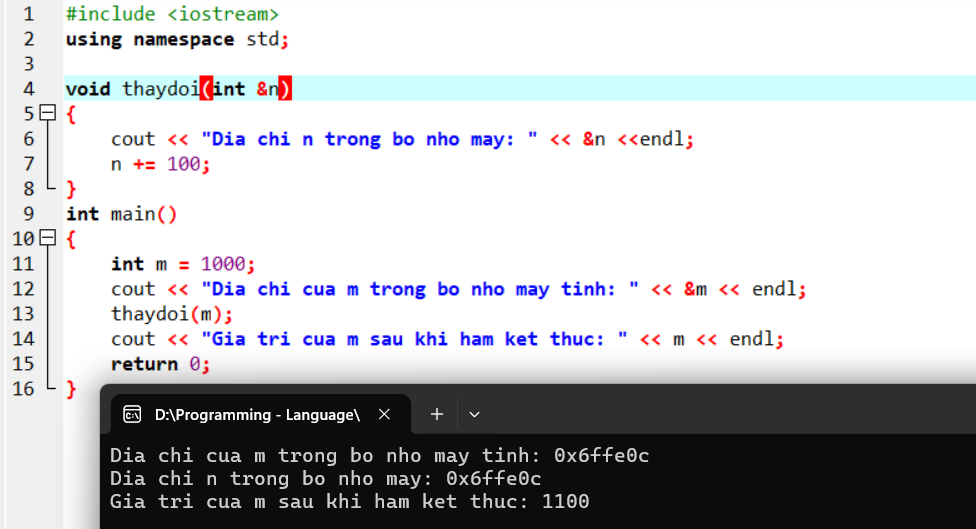
**2. Truyền Tham Chiếu**

**Truyền tham chiếu** - Pass by reference là cách thức xây dựng hàm với đối số là một tham chiếu, có thể hiểu đơn giản thì tham chiếu ở đây là địa chỉ của biến

Để xây dựng tham số là tham chiếu bạn thêm **dấu &** trước tên tham số khi xây dựng hàm.

Hàm khi truyền tham chiếu có thể giúp thay đổi giá trị của đối số sau khi lời gọi hàm kết thúc, hiểu đơn giản đây là phương pháp giúp hàm truy cập trực tiếp vào địa chỉ của đối số và thay đổi giá trị tại ô nhớ đó trong máy tính, dẫn đến giá trị của đối số sẽ bị thay đổi theo.

Xem xét ví dụ sau đây bạn sẽ hiểu rõ hơn về tham chiếu trong C++



1. Hàm thaydoi nhận một tham chiếu đến một biến kiểu int. Điều này có nghĩa là khi hàm được gọi, nó làm việc trực tiếp với biến ban đầu mà không tạo ra một bản sao.

2. cout << "Dia chi n trong bo nho may: " << &n << endl;: In ra địa chỉ bộ nhớ của n.

3. n += 100;: Tăng giá trị của n lên 100. Do n là tham chiếu đến biến gốc, giá trị của biến gốc cũng sẽ thay đổi.

4. int m = 1000;: Khai báo biến m với giá trị ban đầu là 1000.

5. cout << "Dia chi cua m trong bo nho may tinh: " << &m << endl;: In ra địa chỉ bộ nhớ của m.

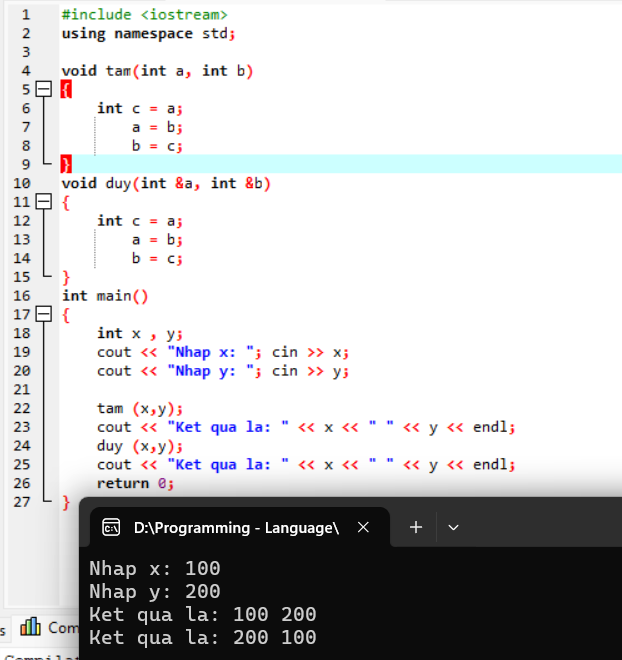
6. thaydoi(m);: Gọi hàm thaydoi với tham số là m. Vì thaydoi nhận tham chiếu, m sẽ được thay đổi bên trong hàm.

7. cout << "Gia tri cua m sau khi ham ket thuc: " << m << endl;: In ra giá trị của m sau khi hàm thaydoi kết thúc.

**Chú ý :**

* Trong C không có truyền tham chiếu, bạn phải sử dụng con trỏ thay vì tham chiếu. Vậy nên nếu bạn đang học lập trình C thì đừng dùng nó một cách vô thức vì thấy người khác cũng dùng nó khi code C, thực tế đây là một hiểu nhầm
* Nếu tham số là tham chiếu thì bạn không thể truyền đối số là giá trị hằng số mà phải thông qua biến, ví dụ trong mã nguồn trên thì bạn không thể gọi thaydoi(1000) vì tham chiếu hoạt động dựa trên địa chỉ của đối số

Ví dụ về việc hoán đổi giá trị của 2 biến thông qua hàm bằng tham chiếu và tham trị :



1. Hàm tam nhận hai tham số a và b theo giá trị. Điều này có nghĩa là các biến x và y khi truyền vào hàm sẽ được sao chép vào a và b. Mọi thay đổi trên a và b không ảnh hưởng đến giá trị thực sự của x và y trong hàm main.

2. Trong hàm, giá trị a và b được hoán đổi, nhưng chỉ thay đổi bản sao, không phải là x và y.

3. Hàm duy nhận hai tham số a và b theo tham chiếu, nghĩa là a và b trỏ đến chính x và y được truyền vào từ main. Bất kỳ thay đổi nào trên a và b sẽ thay đổi trực tiếp giá trị của x và y.

4. Người dùng nhập giá trị cho x và y.

5. Gọi hàm tam(x, y);: Do tam truyền tham số theo giá trị, x và y không thay đổi sau khi gọi hàm này.

6. Gọi hàm duy(x, y);: Do duy truyền tham số theo tham chiếu, x và y bị hoán đổi giá trị.

**Kết quả khi chạy chương trình:**

1. Sau khi gọi hàm tam, giá trị của x và y không thay đổi vì tam chỉ làm việc với bản sao của x và y.
2. Sau khi gọi hàm duy, giá trị của x và y bị hoán đổi vì duy làm việc trực tiếp với x và y thông qua tham chiếu.

**Kết luận bài:** Khi bạn cần xây dựng hàm và muốn những thay đổi trong hàm sẽ được giữ nguyên với đối số sau khi hàm kết thúc thì sử dụng truyền tham chiếu, cũng có một cách khác là sử dụng con trỏ nhưng bạn sẽ học kiến thức này sau.