Question 1

Not answered

Mark 0.00 out of 10.00

[Swap]

Trong bài tập này, bạn sẽ được làm quen với cách *truyền tham chiếu* vào hàm. Trước tiên, hãy xem ví dụ sau:

```
#include <iostream>
using namespace std;

void change(int x) {
    x = x * 2;
}

int main() {
    int a = 5;

    cout << "Gia tri cua a truoc khi goi ham change: " << a << endl;
    change(a);
    cout << "Gia tri cua a sau khi goi ham change:   " << a;

    return 0;
}
```

Hàm change trong chương trình trên được viết với mục đích tăng giá trị của một số nguyên lên hai lần. Tuy nhiên, khi thực thi chương trình, ta nhận được kết quả như sau:

```
Gia tri cua a truoc khi goi ham change: 5
Gia tri cua a sau khi goi ham change:   5
```

Có thể thấy giá trị của biến a trước và sau khi gọi hàm change vẫn được giữ nguyên, mặc dù trong hàm change ta đã có tác động để thay đổi giá trị của tham số truyền vào. Mấu chốt vấn đề ở đây là cách chúng ta truyền tham số cho hàm. Cách mà đoạn code trên truyền tham số vào hàm change được gọi là *truyền giá trị*. Cụ thể, quá trình thực thi của đoạn code trên như sau:

- Khi gọi `change(a)`, giá trị của a (giá trị 5) được truyền vào hàm change;
- Một biến mới `int x` được khởi tạo với giá trị bằng 5;
- Mọi hành vi được sử dụng trong hàm change sau đó chỉ tác động lên biến x, do đó biến a không bị thay đổi gì khi kết thúc hàm change. Kể cả khi ta đặt tên tham số của hàm change là a thay vì x thì mọi thứ vẫn hoạt động như trên, ta có hai biến cùng tên a trong hai hàm change và hàm `main`, tuy nhiên chúng là hai thực thể khác nhau.

Để giải quyết vấn đề này, hàm change cần sử dụng một kiểu biến khác, đó là *biến tham chiếu*. Biến tham chiếu có thể được hiểu như "đại diện" của một biến khác. Khi sử dụng biến tham chiếu, ta sẽ truy cập trực tiếp vào vùng nhớ của biến được "đại diện", tức là mọi thay đổi trên biến tham chiếu cũng là những thay đổi trên biến được "đại diện". Để khai báo biến tham chiếu, ta chỉ cần thêm dấu `&` ở trước tên biến và sau kiểu dữ liệu. Ví dụ:

```
double a = 4.18;
double &x = a; // x = 4.18
x = 5.12;      // a = 5.12
a += 1;        // a = 6.12
               // x = 6.12
```

Như vậy, hàm change sẽ được sửa lại như sau:

```
void change(int &x) {
    x = x * 2;
}
```

Kết quả chạy chương trình sau khi sửa hàm change:

```
Gia tri cua a truoc khi goi ham change: 5
Gia tri cua a sau khi goi ham change:   10
```

Bài tập

Viết hàm `void swap(int &a, int &b)` để tráo đổi giá trị của hai biến kiểu nguyên được truyền vào hàm.

Đầu vào

Đầu vào từ bàn phím, gồm hai số nguyên a và b phân tách nhau bởi một dấu cách.

Đầu ra

In ra màn hình giá trị của a và b sau khi gọi hàm `swap(a, b)`. Giá trị của a và b nằm trên một dòng, phân tách nhau bởi một dấu cách.

Lưu ý

Các phần nhập/xuất dữ liệu trong hàm `main` đã được viết sẵn cho bạn. Bạn chỉ cần viết định nghĩa hàm `swap` tại vị trí được yêu cầu trong ô trả lời.

For example:

Input	Result
1 2	2 1

Answer:

Reset answer

```cpp
#include <iostream>
using namespace std;

void swap(int &a, int &b) {
    // complete the function

}

int main() {
    int a, b;
    cin >> a >> b;
    swap(a, b);
    cout << a << " " << b;

    return 0;
}
```

	Input	Expected	Got	
✓	1 2	2 1	2 1	✓
✓	10 20	20 10	20 10	✓

Passed all tests! ✓

▸ SHOW/HIDE QUESTION AUTHOR'S SOLUTION (CPP)

Back to Course