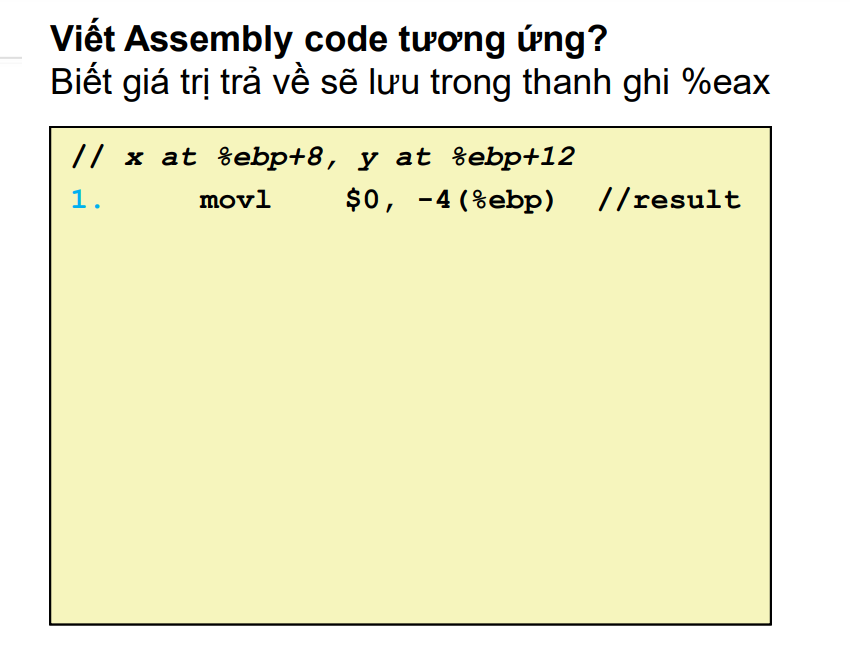
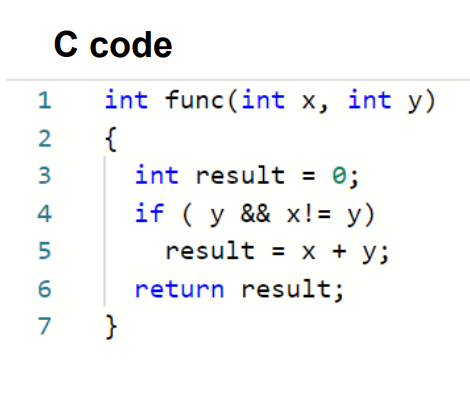
**Bài tập: Chuyển mã rẽ nhanh có điều kiện**

Bài 3:





// x at %ebp+8, y at %ebp+12

1. movl $0, -4(%ebp) // Result

2. cmpl $0, 12(%ebp) // So sánh y với zero

3. je .L2 // Nhảy đến L2 nếu như y == 0

4. cmpl 8(%ebp), 12(%ebp) // So sánh x và y

5. je .L2 // Nhảy đến L2 nếu như x==y

6. movl 8(%ebp), %eax // Đẩy x vào thanh ghi %eax

7. addl 12(%ebp), %eax // Lấy giá trị trên bộ nhớ y công thanh ghi %eax rồi lưu vào %eax

8. movl %eax, -4(%ebp) // tải giá trị %eax vào ô nhớ của result

9. .L2

10. movl -4(%ebp), %eax // Tải giá trị trả về vào lại %eax

Bài 4:

// a at %ebp+8, b at %ebp+12, c at %ebp+16

1. movl $0, -4(%ebp) //sum

2. cmpl $0, 16(%ebp) //so sánh c với Zero

3. jge .L2 // nếu c >= 0 nhảy tới .L2

4. movl 8(%ebp), %eax // Đưa a vào thanh ghi %eax

5. andl 12(%ebp), %eax // a & b

6. xorl 16(%ebp), %eax // (a & b) ^ c

7. movl %eax, -4(%ebp) // Chuyển kết quả về lưu tại thanh ghi %eax -> tải giá trị %eax vào ô nhớ của sum

8. cmpl 8(%ebp), 12(%ebp) // So sánh a và b

9. jne .L2 // Nếu a != b nhảy tới .L2

10. movl 8(%ebp), %eax // Đưa a vào thanh ghi %eax

11. andl 12(%ebp), %eax // Tính a & b

12. xorl 16(%ebp), %eax // Tính (a & b) ^ c

13. movl %eax, -4(%ebp) // Chuyển kết quả về lưu tại thanh ghi %eax -> tải giá trị %eax vào ô nhớ của sum

1. .L2
2. movl -4(%ebp), %eax // Tải giá trị trả về vào %eax