

```
#include <stdio.h>
int main() {
    int a = 10, b = 20, c = 30, d;
    d = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
    printf("%d", d);
    return 0;
}
```


Giải thích từng phần

1. Toán tử điều kiện bậc ba `? : :`

- Toán tử này có cú pháp là `condition ? expr_if_true : expr_if_false`.
- Nếu `condition` là đúng (true), biểu thức `expr_if_true` sẽ được đánh giá và trả về; nếu `condition` sai (false), biểu thức `expr_if_false` sẽ được đánh giá và trả về.

2. Biểu thức trong mã này:

c

 Sao chép mã

```
d = (a > b) ? (a > c ? a : c) : (b > c ? b : c);
```

Biểu thức này có hai cấp độ của toán tử điều kiện bậc ba để kiểm tra xem `a`, `b`, hay `c` là lớn nhất:

- **Bước đầu tiên:** `(a > b) ? ... : ...`
 - Điều kiện `(a > b)` kiểm tra xem `a` có lớn hơn `b` hay không.
 - Nếu đúng `(a > b)`, nó sẽ tiếp tục với biểu thức `(a > c ? a : c)`.
 - Nếu sai `(a <= b)`, nó sẽ tiếp tục với biểu thức `(b > c ? b : c)`.
- **Bước thứ hai:**
 - Nếu `(a > b)` là đúng, biểu thức `(a > c ? a : c)` sẽ được đánh giá:
 - Kiểm tra xem `a` có lớn hơn `c` hay không.
 - Nếu đúng, giá trị của `a` sẽ được gán cho `d`; nếu sai, giá trị của `c` sẽ được gán cho `d`.
 - Nếu `(a > b)` là sai, biểu thức `(b > c ? b : c)` sẽ được đánh giá:
 - Kiểm tra xem `b` có lớn hơn `c` hay không.
 - Nếu đúng, giá trị của `b` sẽ được gán cho `d`; nếu sai, giá trị của `c` sẽ được gán cho `d`.

Phân tích với các giá trị cụ thể

Trong chương trình, ta có:

- `a = 10`
- `b = 20`
- `c = 30`

Áp dụng biểu thức `d = (a > b) ? (a > c ? a : c) : (b > c ? b : c);`:

1. Đánh giá `(a > b)`:

- `10 > 20` là sai.
- Do đó, biểu thức `b > c ? b : c` sẽ được đánh giá.

2. Đánh giá `(b > c ? b : c)`:

- `20 > 30` là sai.
- Vì vậy, `c` sẽ được gán cho `d`.


Kết quả là `d = 30`.



Kết quả đầu ra

Chương trình sẽ in ra:

30

 Sao chép mã

Tóm tắt

Biểu thức điều kiện bậc ba trong đoạn mã này thực hiện việc so sánh ba biến `a`, `b`, và `c` để tìm giá trị lớn nhất. Toán tử điều kiện kiểm tra từng điều kiện một và trả về giá trị phù hợp, đảm bảo rằng `d` sẽ chứa giá trị lớn nhất trong ba số.