

Trong đoạn mã này, macro `FUN(i, j)` sử dụng toán tử `##` để nối hai tham số `i` và `j` lại với nhau thành một tên duy nhất. Hãy phân tích từng phần chi tiết.

```
c Sao chép mã

#include<stdio.h>
#define FUN(i, j) i##j

int main()
{
    int va1 = 10;
    int va12 = 20;
    printf("%d\n", FUN(va1, 2));
    return 0;
}
```

Phân tích từng phần

1. **Macro `#define FUN(i, j) i##j`:

- `#define` là từ khóa để khai báo một macro trong C.
- `FUN(i, j)` là một macro nhận hai tham số `i` và `j`.
- Toán tử `##` trong macro được gọi là **toán tử ghép nối (token-pasting operator)**, có nhiệm vụ nối hai tham số lại thành một mã định danh (identifier).
- Trong trường hợp này, `i##j` sẽ nối `i` và `j` lại thành một tên biến duy nhất khi sử dụng macro.

2. **Sử dụng `FUN(va1, 2)` trong `printf`:

- Khi `FUN(va1, 2)` được gọi trong `printf`, trình biên dịch sẽ thay `FUN(va1, 2)` bằng `va1##2`.
- Toán tử `##` sẽ nối `va1` và `2` lại thành `va12`.
- Kết quả là `FUN(va1, 2)` sẽ được thay thế bằng `va12` trong mã.

3. **Giá trị của `va12`:

- Trong `main`, ta đã khai báo `int va12 = 20;`, vì vậy `va12` có giá trị là `20`.

4. **Kết quả của `printf("%d\n", FUN(va1, 2));`:

- Sau khi thay thế, câu lệnh `printf` sẽ trở thành `printf("%d\n", va12);`.

- Do đó, chương trình sẽ in ra giá trị của `va12`, tức là `20`.

Kết quả đầu ra

Khi chạy chương trình, kết quả sẽ là:

...

20

...

Tóm tắt

Toán tử `##` trong macro `FUN(i, j)` nối hai tham số `i` và `j` thành một mã định danh duy nhất. Trong trường hợp này, `FUN(va1, 2)` sẽ trở thành `va12`, và giá trị của `va12` là `20`, nên chương trình sẽ in ra `20`.