

Khi nào dùng capso() (chỉ gọi hàm thôi) ?

Ví dụ:

```
cpp Sao chép mã  
  
capso(n, a, k);
```

=> Trường hợp này chỉ hợp lý nếu **hàm capso không có return, mà chỉ in ra hoặc xử lý trực tiếp.**

Tức là hàm có kiểu `void`.

Ví dụ:

```
cpp Sao chép mã  
  
void capso(int n, int a[], int k) {  
    int count = 0;  
    for (int i = 0; i < n; i++) {  
        for (int j = i + 1; j < n; j++) {  
            if (a[i] + a[j] == k) {  
                count++;  
            }  
        }  
    }  
    cout << "Số cặp thỏa mãn: " << count << endl;  
}
```

Trong `main` chỉ cần:

```
cpp Sao chép mã  
  
capso(n, a, k); // gọi thôi, không cần cout
```

2. Khi nào dùng cout << capso(...)

Ví dụ:

```
cpp Sao chép mã  
  
cout << capso(n, a, k);
```

=> Trường hợp này hợp lý nếu **hàm capso trả về một giá trị (int, long long, ...).**


Tức là hàm có kiểu **không phải** `void` (ví dụ `int capso(...)`).

Khi đó `capso(...)` được coi như **một con số** mà mình in ra luôn.

3. Khi nào dùng `int result = capso(...)`

Ví dụ:

cpp

 Sao chép mã

```
int result = capso(n, a, k);  
cout << result;
```

=> Dùng khi mình muốn **lưu kết quả vào biến để dùng nhiều lần** (không chỉ in ra).