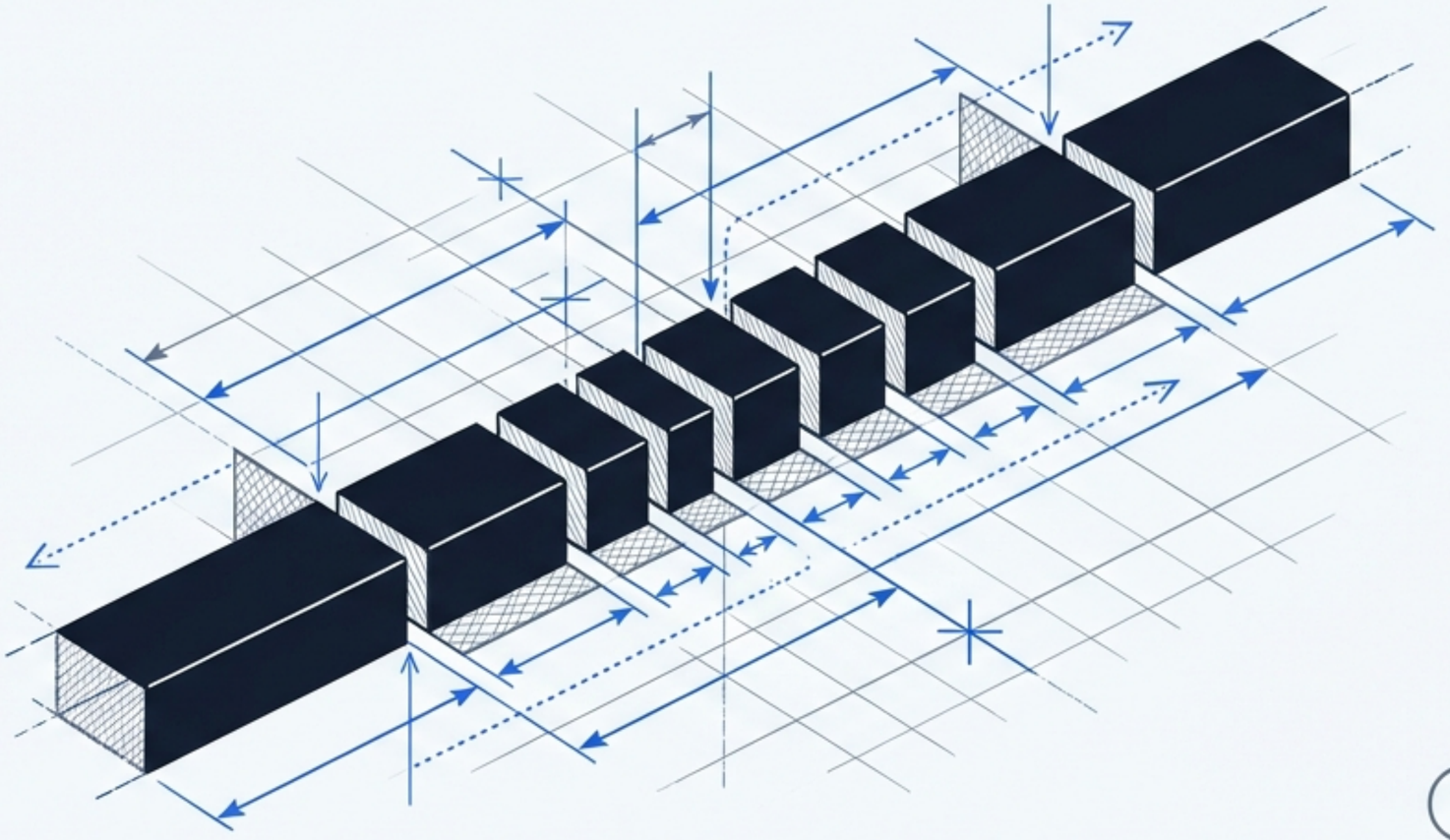
Giải Mã C: Thuật Toán Đếm Tần Suất Từ

Phân tích từ trong chuỗi không chỉ là viết code, mà là cách chúng ta tổ chức dữ liệu và tư duy thuật toán.

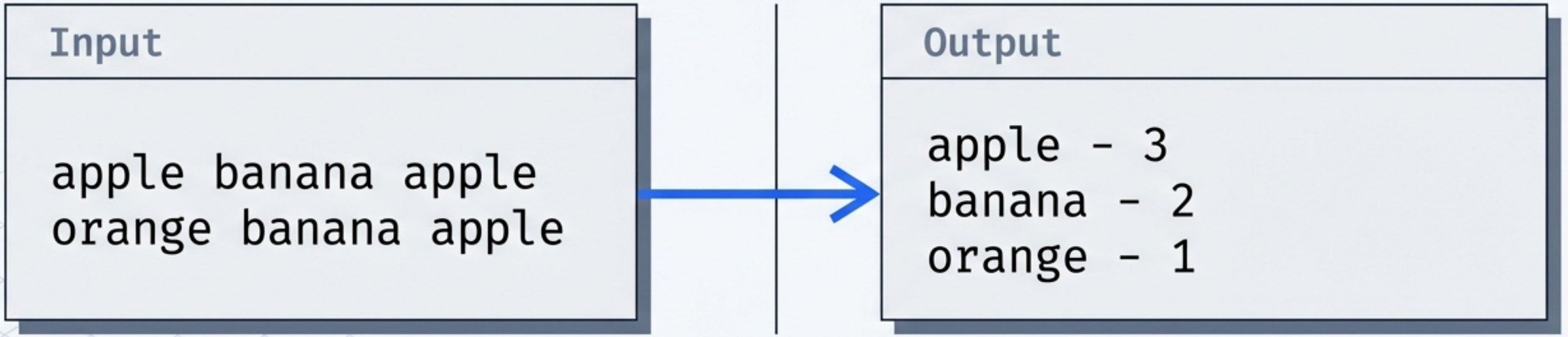


Error404-Labs.Info.Vn

Xác Định Rõ Yêu Cầu Bài Toán

Một “từ” được định nghĩa là một chuỗi ký tự liên tiếp không chứa khoảng trắng.
Mục tiêu đầu ra:

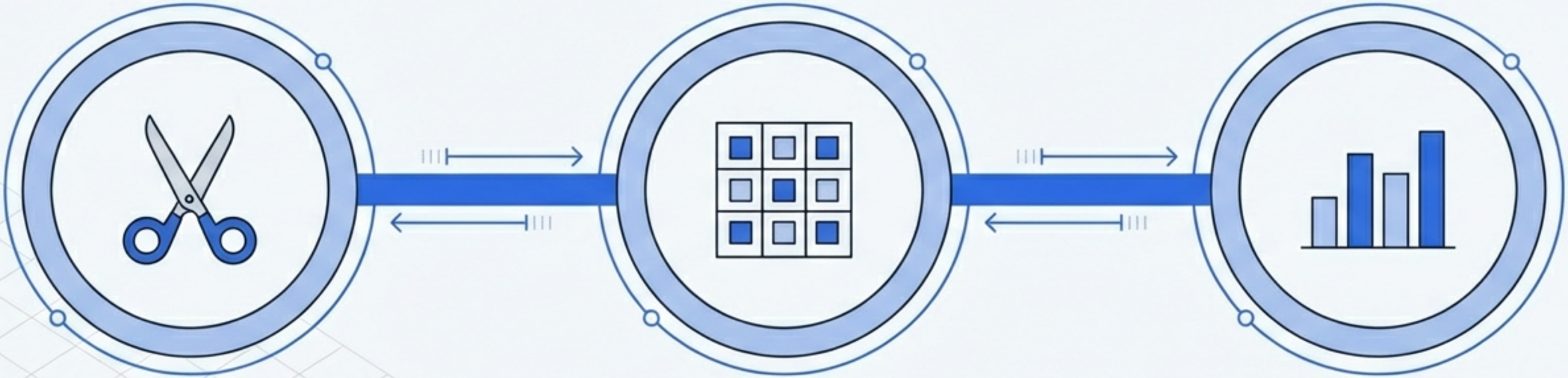
1. Trích xuất danh sách các từ đã tách.
2. Thống kê số lần xuất hiện của từng từ.



Error404-Labs.Info.Vn

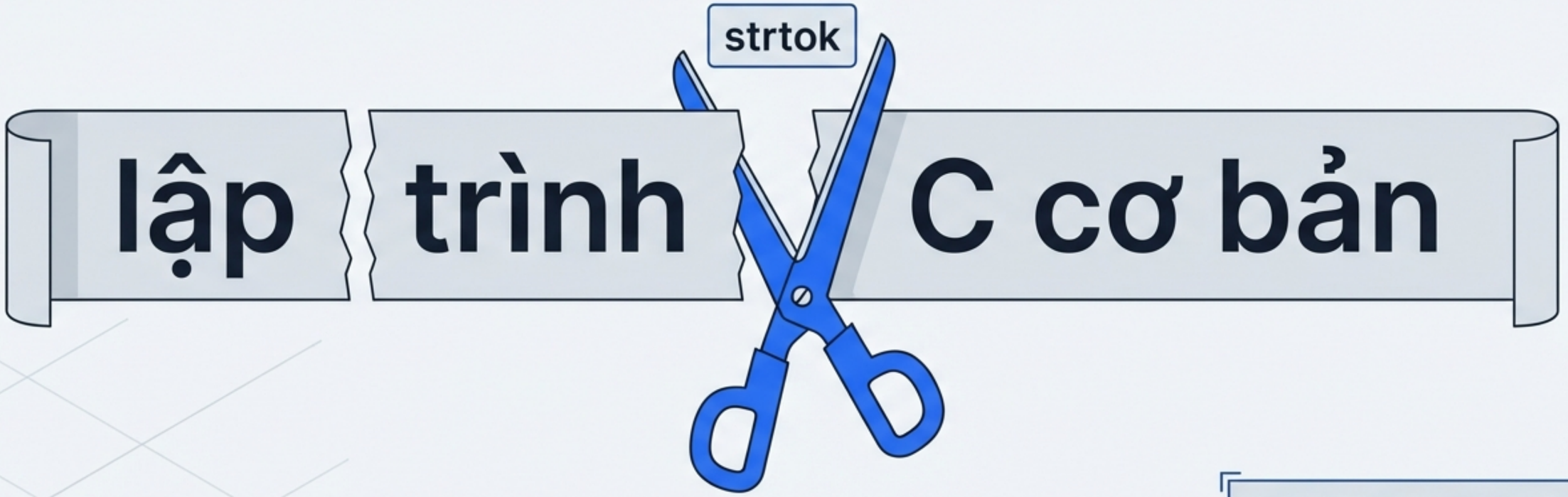
Ý Tưởng Giải Quyết

Kiến trúc thuật toán được chia thành 3 bước độc lập:
Bước 1: Tách (Split) - Cắt chuỗi thành các từ riêng biệt.
Bước 2: Lưu (Store) - Đưa các từ vào cấu trúc dữ liệu mảng.
Bước 3: Đếm (Count) - Duyệt và thống kê tần suất xuất hiện.



Bước 1: Nguyên Lý Cắt Chuỗi

Hàm strtok hoạt động như một công cụ chia tách, tìm kiếm các khoảng trắng (" ") và tách chuỗi ban đầu thành các phần nhỏ hơn gọi là token.



Error404-Labs.Info.Vn

Tách Chuỗi Bằng strtok

Error404-Labs.Info.Vn

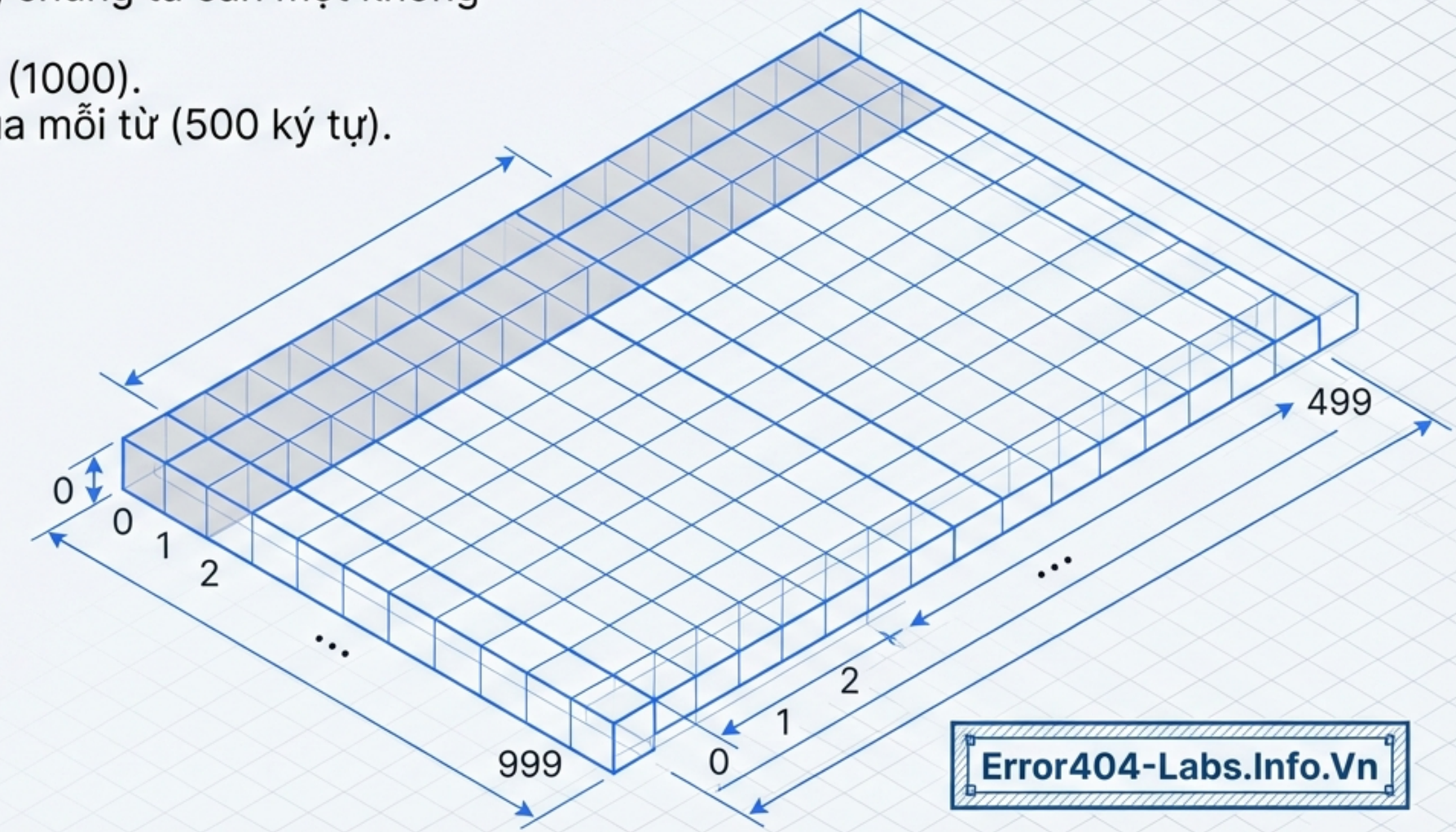
Khởi tạo lần đầu truyền chuỗi text. Các lần gọi tiếp theo truyền NULL để tiếp tục quá trình tách.

```c
char *token = strtok(text, " ");

// ... trong vòng lặp ...

token = strtok(NULL, " ");
```

Cảnh báo: Hàm này thay đổi chuỗi gốc bằng cách thay thế dấu cách bằng ký tự '\0'.

Bước 2: Xây Dựng Cấu Trúc Lưu Trữ

Để lưu trữ danh sách các từ, chúng ta cần một không gian hai chiều.
Trục dọc: Số lượng từ tối đa (1000).
Trục ngang: Độ dài tối đa của mỗi từ (500 ký tự).

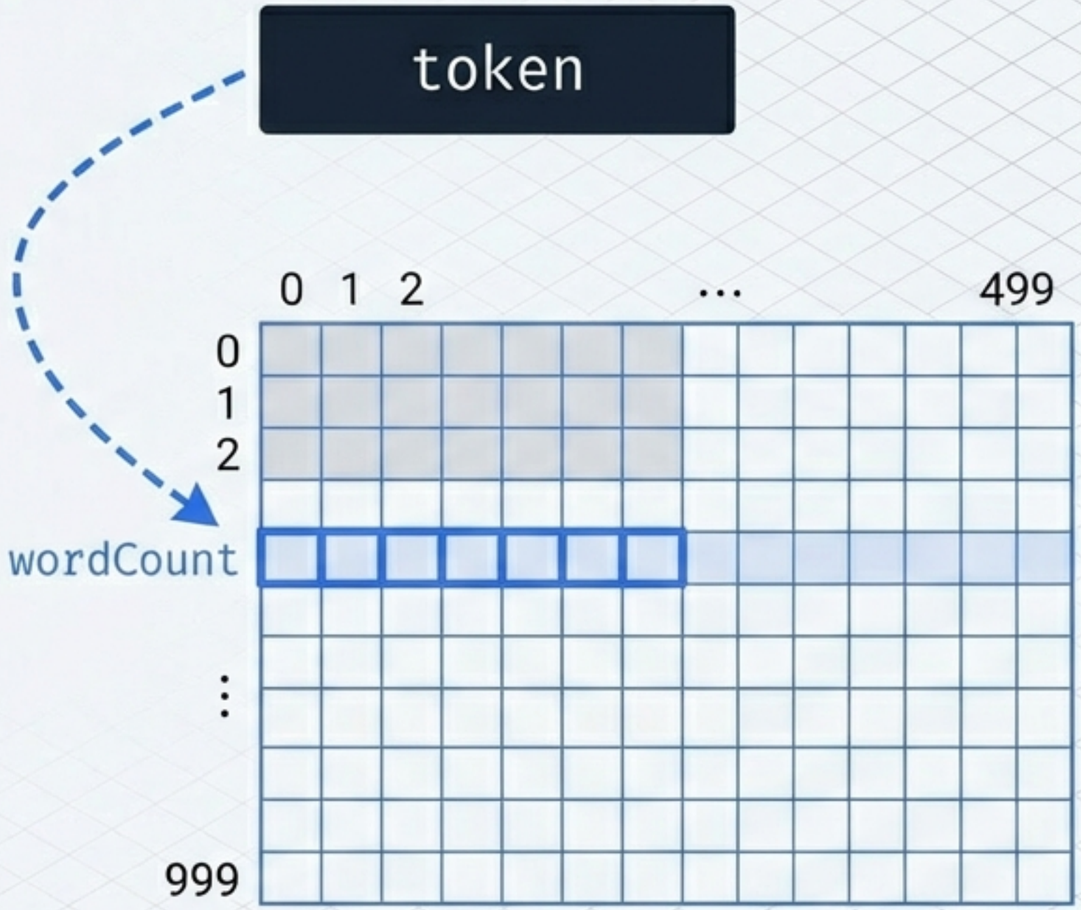


Error404-Labs.Info.Vn

Đưa Dữ Liệu Vào Mảng 2 Chiều

Khai báo mảng: char words[1000][500];
Sử dụng hàm strcpy để sao chép từng token đã tách vào các hàng liên tiếp của mảng words.
Biến wordCount vừa làm chỉ số mảng, vừa đóng vai trò đếm tổng số từ.

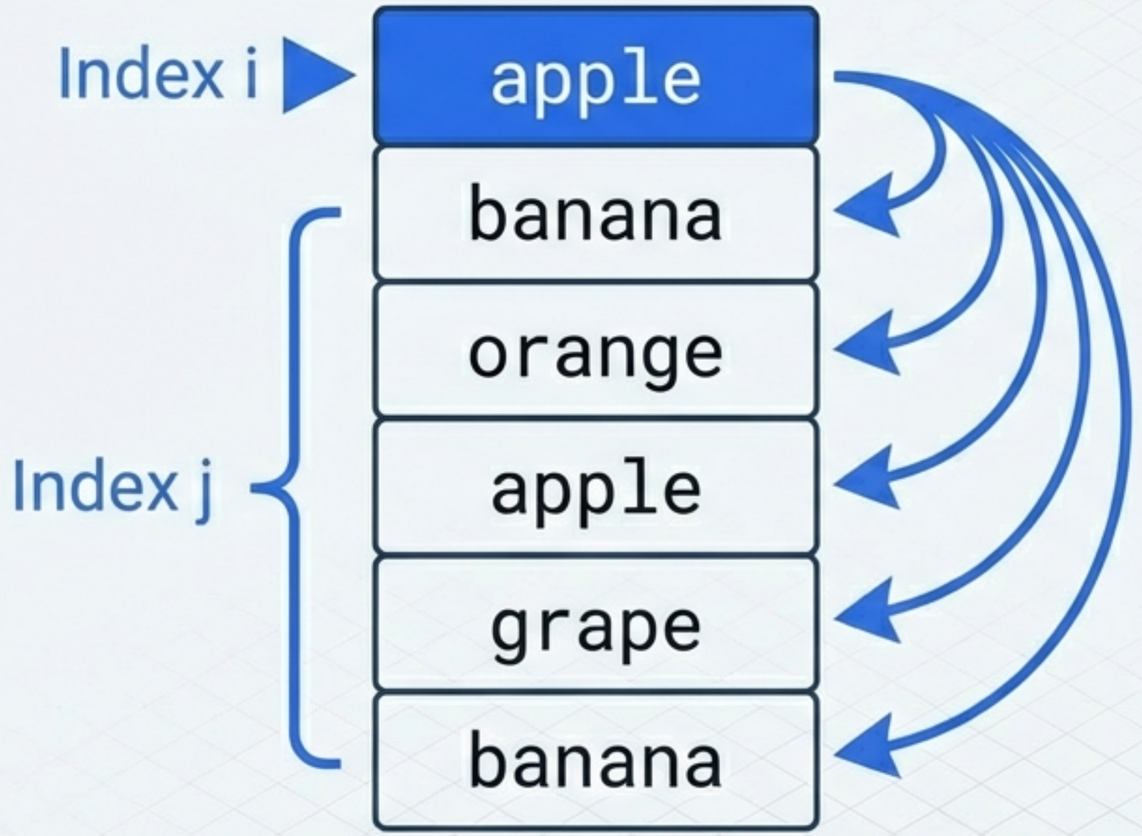
```
strcpy(words[wordCount++], token);
```



Error404-Labs.Info.Vn

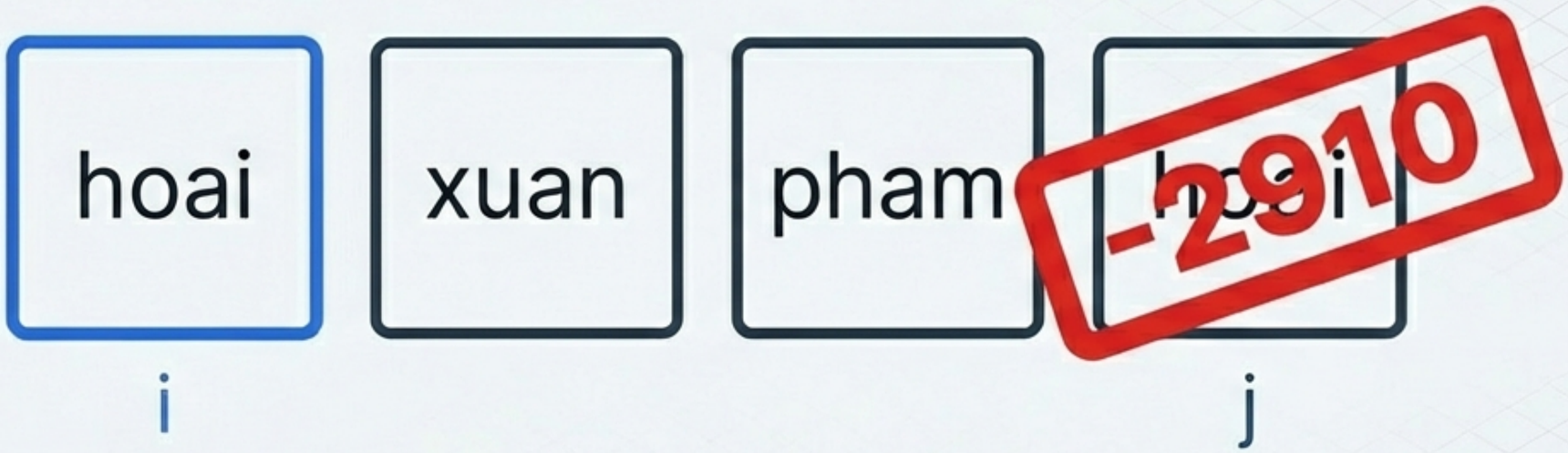
Bước 3: Bài Toán Đếm Tần Suất

Với mỗi từ (vị trí i), chúng ta cần duyệt qua toàn bộ các từ đứng sau nó (vị trí j) để kiểm tra xem có sự trùng lặp hay không. Đây là nền tảng của tư duy thuật toán duyệt lồng nhau $O(n^2)$.



Kỹ Thuật Gắn Cờ - Tránh Đếm Trùng

Làm sao để không đếm một từ hai lần?
Nếu phát hiện từ ở vị trí j giống với từ đang xét ở vị trí i, ta đánh dấu từ ở vị trí j bằng một giá trị đặc biệt: -2910.
Khi vòng lặp chính gặp giá trị -2910, nó sẽ bỏ qua (continue) vì từ này đã được thống kê.



Error404-Labs.Info.Vn

Thực Thi Logic Bằng Code

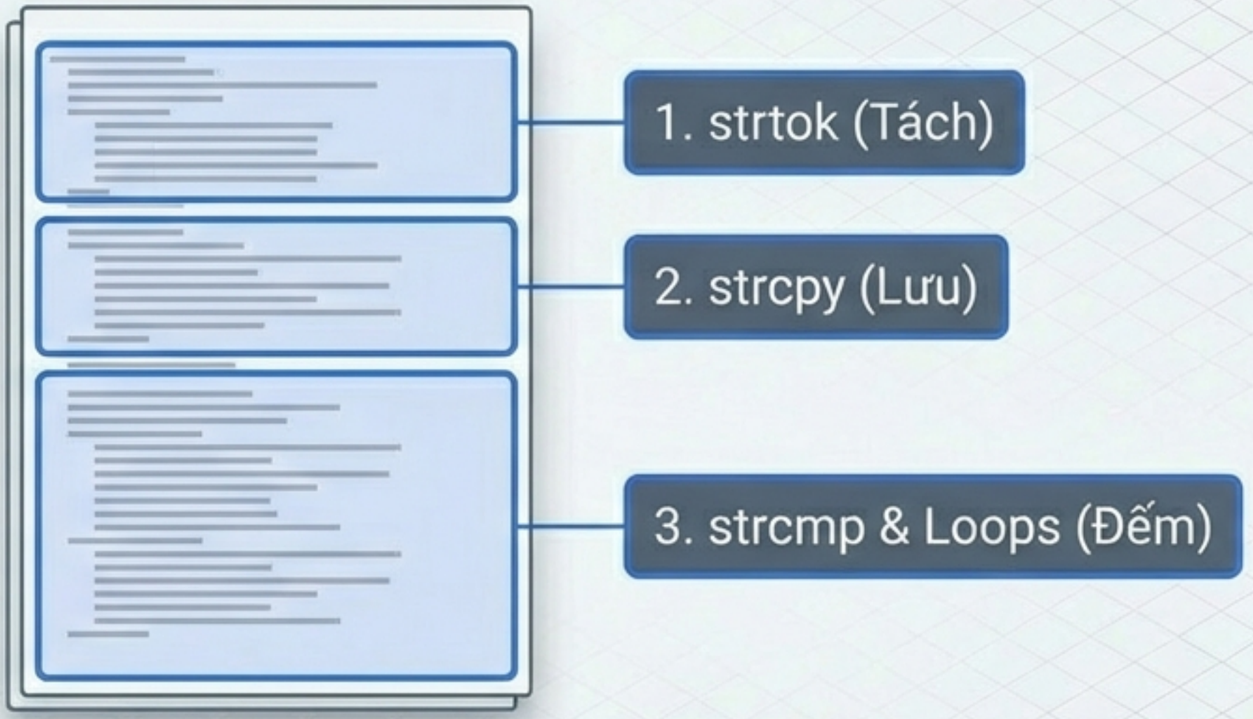
Sử dụng memset để khởi tạo mảng tần suất freq bằng 0. Sử dụng vòng lặp lồng nhau kết hợp strcmp để so sánh và biến cờ -2910 để đánh dấu.

```
    if(strcmp(words[i], words[j]) == 0) {
        freq[i]++;
        freq[j] = -2910;    // Đánh dấu đã đếm
    }
```

Error404-Labs.Info.Vn

Kiến Trúc Code Tổng Thể

Toàn bộ hệ thống hoạt động mạch lạc theo 3 khối logic đã phân tích. Sự rõ ràng trong phân chia luồng xử lý giúp code dễ bảo trì và nâng cấp.



Error404-Labs.Info.Vn

Ví Dụ Chạy Thử (Live Run)

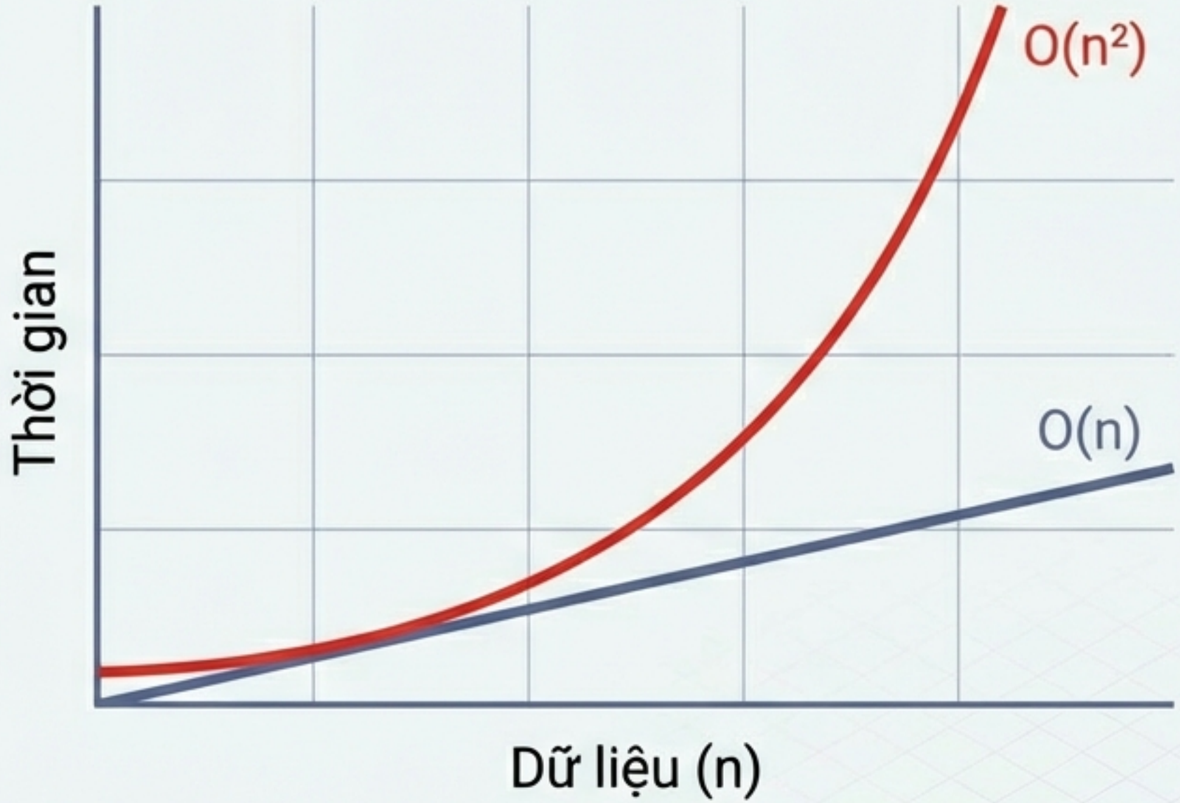
Quá trình xử lý: Thuật toán nhận diện sự lặp lại của "pham" và "hoai", đồng thời áp dụng cờ -2910 để bỏ qua các lần xuất hiện thứ hai.

```
Input:
pham xuan hoai pham hoai

Output:
pham - 2
xuan - 1
hoai - 2
```


Error404-Labs.Info.Vn

Phân Tích Độ Phức Tạp Thuật Toán

Cấu trúc vòng lặp lồng nhau mang lại độ phức tạp thời gian là $O(n^2)$.
Cách tiếp cận này trực quan, hoàn hảo cho tư duy học thuật và bài tập sinh viên.
Để xử' dữ liệu lớn hơn, có thể tối ưu bằng cách sắp xếp (sort) mảng trước khi đếm.



Mẹo Tối Ưu: Sắp xếp (Sorting) mảng trước sẽ loại bỏ nhu cầu duyệt $O(n^2)$.

Bộ Công Cụ Thao Tác Chuỗi C

Làm chủ việc thao tác chuỗi trong C đòi hỏi sự hiểu biết sâu sắc về các hàm cốt lõi. Trong thuật toán này, chúng ta đã ứng dụng thành thạo:

strtok() Chia tách chuỗi qua delimiter.	**strcpy()** Sao chép dữ liệu chuỗi an toàn vào mảng.
strcmp() So sánh giá trị hai chuỗi ký tự.	**memset()** Khởi tạo giá trị mặc định cho mảng bộ nhớ.

Error404-Labs.Info.Vn