

- Prompt: *“So sánh Array và Linked List theo tốc độ truy cập, bộ nhớ, tình huống áp dụng. Sau đó đưa code minh họa thao tác chèn phần tử trong mỗi cấu trúc.”*
- Nếu AI chỉ trả lời lý thuyết mà chưa có code → chỉnh Prompt để bổ sung.

Tiêu chí	Array	Linked list
Cấu trúc Bộ nhớ	Các phần tử được lưu trữ ở các vị trí bộ nhớ liên kề.	Các phần tử được lưu trữ ở các vị trí bộ nhớ không liên kề, liên kết với nhau bằng con trỏ.
Tốc độ Truy cập	Nhanh. Có thể truy cập bất kỳ phần tử nào ngay lập tức bằng chỉ số (index) vì vị trí đã biết.	Chậm. Phải duyệt tuần tự từ đầu đến phần tử mong muốn.
Tốc độ Chèn/Xóa	Chậm. Khi chèn/xóa ở giữa, cần phải dịch chuyển tất cả các phần tử còn lại để duy trì tính liên kề.	Nhanh. Chỉ cần thay đổi con trỏ của các nút lân cận.
Bộ nhớ	Bộ nhớ tĩnh. Kích thước cố định (hoặc cần cấp phát lại toàn bộ khi đầy), dễ gây lãng phí bộ nhớ nếu không dùng hết.	Bộ nhớ linh hoạt. Kích thước thay đổi động theo nhu cầu. Tuy nhiên, tốn thêm bộ nhớ cho mỗi phần tử để lưu trữ con trỏ.

Tình huống Áp dụng	Khi cần truy cập ngẫu nhiên thường xuyên và kích thước dữ liệu ít thay đổi.	Khi cần chèn/xóa thường xuyên và vị trí chính xác không quan trọng.
--------------------	---	---

🌟 A. Chèn phần tử vào Array

👉 Chèn số `x` vào vị trí `pos`

```
c
#include <stdio.h>

int main() {
    int arr[10] = {1, 2, 3, 4, 5};
    int n = 5;
    int x = 99;
    int pos = 2;

    // Dời phần tử sang phải
    for (int i = n; i > pos; i--) {
        arr[i] = arr[i - 1];
    }

    arr[pos] = x;
    n++;

    // In mảng
    for (int i = 0; i < n; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```

✦ B. Chèn phần tử vào Linked List

👉 Chèn node mới sau node đầu

```
c Copy code

#include <stdio.h>
#include <stdlib.h>

struct Node {
    int data;
    struct Node* next;
};

int main() {
    // Tạo node đầu
    struct Node* head = (struct Node*)malloc(sizeof(struct Node));
    head->data = 1;
    head->next = NULL;

    // Tạo node mới
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = 99;

    // Chèn sau head
    newNode->next = head->next;
    head->next = newNode;

    // In danh sách
    struct Node* temp = head;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL");

    return 0;
}
```