

F. Print Even Indices

Huấn luyện viên tư duy thuật toán

Micro-Chunks Learning

Ngày 16 tháng 1 năm 2026

Bước 1: Tiếp nhận & Phẫu thuật (Briefing)

Tóm tắt bài toán

Bạn có một danh sách các số. Nhiệm vụ của bạn là:

- ❶ Chỉ nhìn vào những vị trí (chỉ số) **chẵn** (0, 2, 4...).
- ❷ In những số đó theo **thứ tự ngược lại** (từ cuối lên đầu).
- ❸ **Điều kiện bắt buộc**: Phải dùng **Đệ quy (Recursion)**.

Lộ trình tư duy (Chunks)

- **Chunk 1**: Xác định vị trí chẵn.
- **Chunk 2**: Hiểu về Đệ quy - "Búp bê Nga".
- **Chunk 3**: Xử lý thứ tự ngược & Điểm dừng.
- **Chunk 4**: Tổng kết thuật toán.

Chunk 1: Xác định vị trí chẵn (Even Indices)

- Mảng là một dãy các ngăn tủ đánh số từ 0 đến $N - 1$.
- Ví dụ với $N = 4$, mảng: 1 4 2 7
 - Ngăn 0: số 1 (**Chẵn**)
 - Ngăn 1: số 4 (**Lẻ**)
 - Ngăn 2: số 2 (**Chẵn**)
 - Ngăn 3: số 7 (**Lẻ**)

Quy luật

Để nhảy từ vị trí chẵn này sang vị trí chẵn tiếp theo, chúng ta thực hiện bước nhảy: `index = index + 2`.

Thử thách tư duy

Nếu mảng có 7 phần tử (0 đến 6), các chỉ số cần quan tâm là gì?

Chunk 1: Xác định vị trí chẵn (Even Indices)

- Mảng là một dãy các ngăn tử đánh số từ 0 đến $N - 1$.
- Ví dụ với $N = 4$, mảng: 1 4 2 7
 - Ngăn 0: số 1 (**Chẵn**)
 - Ngăn 1: số 4 (**Lẻ**)
 - Ngăn 2: số 2 (**Chẵn**)
 - Ngăn 3: số 7 (**Lẻ**)

Quy luật

Để nhảy từ vị trí chẵn này sang vị trí chẵn tiếp theo, chúng ta thực hiện bước nhảy: `index = index + 2`.

Thử thách tư duy

Nếu mảng có 7 phần tử (0 đến 6), các chỉ số cần quan tâm là gì? **Đáp án:** 0, 2, 4, và 6.

Chunk 2: Đệ quy – "Hành động lùi lại"

Đệ quy giống như mở búp bê Nga: Mở con lớn \rightarrow gặp con nhỏ hơn \rightarrow ...
 \rightarrow **Base Case** (con nhỏ nhất).

Cơ chế hoạt động

- ➊ **Đi tiếp:** Gọi chính nó với chỉ số tiếp theo (`index + 2`).
- ➋ **Quay về & Thực hiện:** In ra màn hình khi quay ngược trở lại.

Thử thách: Đặt lệnh `cout` « `A[i]` ở đâu?

- Đặt **trước** khi gọi hàm đệ quy tiếp theo.
- Đặt **sau** khi gọi hàm đệ quy tiếp theo.

Chunk 2: Đệ quy – "Hành động lùi lại"

Đệ quy giống như mở búp bê Nga: Mở con lớn \rightarrow gắp con nhỏ hơn \rightarrow ...
 \rightarrow **Base Case** (con nhỏ nhất).

Cơ chế hoạt động

- ➊ **Di tiếp:** Gọi chính nó với chỉ số tiếp theo ($\text{index} + 2$).
- ➋ **Quay về & Thực hiện:** In ra màn hình khi quay ngược trở lại.

Thử thách: Đặt lệnh `cout << A[i]` ở đâu?

A. Đặt **trước** khi gọi hàm đệ quy tiếp theo.

B. Đặt **sau** khi gọi hàm đệ quy tiếp theo.

Đáp án B: Để chờ các hàm sau chạy xong (chạm đến cuối mảng) rồi mới in ngược lại.

Chunk 3: Điểm dừng (Base Case)

Tại sao cần điểm dừng?

Để tránh lỗi **Stack Overflow** (Tràn bộ nhớ) do gọi hàm vô tận.

Thử thách tư duy

Điều kiện dừng nào an toàn nhất cho mảng có N phần tử?

- A. Khi `index < N`.
- B. Khi `index >= N`.
- C. Khi `index == N`.

Chunk 3: Điểm dừng (Base Case)

Tại sao cần điểm dừng?

Để tránh lỗi **Stack Overflow** (Tràn bộ nhớ) do gọi hàm vô tận.

Thử thách tư duy

Điều kiện dừng nào an toàn nhất cho mảng có N phần tử?

- A. Khi $\text{index} < N$.
- B. Khi $\text{index} \geq N$.
- C. Khi $\text{index} == N$.

Đáp án B: $\text{index} \geq N$ đảm bảo chúng ta dừng lại ngay khi vượt quá phạm vi mảng.

Chunk 4: Luồng hoạt động của thuật toán

Giả sử $N = 3$, mảng có chỉ số 0, 2:

- `printEven(0)` gọi `printEven(2)`
- `printEven(2)` gọi `printEven(4)`
- `printEven(4)` thỏa mãn `index >= 3` → **Dừng**.
- Quay về `printEven(2)`: In `A[2]`.
- Quay về `printEven(0)`: In `A[0]`.

Câu hỏi khởi động

Trong hàm `main`, ta gọi hàm lần đầu với tham số nào?

Chunk 4: Luồng hoạt động của thuật toán

Giả sử $N = 3$, mảng có chỉ số 0, 2:

- `printEven(0)` gọi `printEven(2)`
- `printEven(2)` gọi `printEven(4)`
- `printEven(4)` thỏa mãn `index >= 3` → **Dừng**.
- Quay về `printEven(2)`: In `A[2]`.
- Quay về `printEven(0)`: In `A[0]`.

Câu hỏi khởi động

Trong hàm `main`, ta gọi hàm lần đầu với tham số nào? **Đáp án:**
`printEven(0, N, A)`.

Cấu trúc mã nguồn C++

```
1 void printEven(int index, int N, long long A[]) {  
2     // 1. Điểm dừng (Base Case)  
3     if (index >= N) {  
4         return;  
5     }  
6  
7     // 2. Lỗi gọi đệ quy (Bước nhảy)  
8     // Đi sau vào các chỉ số tiếp theo trước  
9     printEven(index + 2, N, A);  
10  
11    // 3. Hành động (In ra)  
12    // Khi hàm quay ngược trở lại, ta mới in giá trị  
13    cout << A[index] << " ";  
14 }
```

Lưu ý kỹ thuật

- **Kiểu dữ liệu:** Dùng long long cho giá trị mảng.
- **Giới hạn:** N lên tới 10^3 , mảng khai báo A[1005].

Tổng kết & Bước tiếp theo

Bạn đã nắm vững:

- Cách nhảy bước đôi trong mảng.
- Cơ chế Stack của đệ quy để đảo ngược dữ liệu.
- Cách xác định điều kiện dừng an toàn.

Bước tiếp theo

1. Thử tự viết code hoàn chỉnh dựa trên cấu trúc trên.
2. Tìm hiểu cách truyền mảng/vector tối ưu vào hàm.