

Codeforces 110A - Nearly Lucky Number

Rèn luyện kỹ năng đọc hiểu và cài đặt (Implementation)

Slide Learning CPP

Ngày 14 tháng 1 năm 2026

Mục tiêu bài học

Hôm nay chúng ta sẽ cùng "mổ xẻ" bài toán **Codeforces 110A - Nearly Lucky Number**. Đây là bài toán tuyệt vời để:

- Rèn luyện kỹ năng đọc hiểu định nghĩa.
- Tránh các bẫy tư duy lỗi mòn khi xử lý số học.

Bước 1: Tiếp nhận & Phẫu thuật (Briefing)

Hãy nhìn vào lỗi vấn đề dưới ngôn ngữ "người thường":

Định nghĩa 1: Số May Mắn (Lucky Number)

Là số dương mà trong biểu diễn thập phân **CHỈ** chứa các chữ số **4** và **7**.

- **Đúng:** 4, 7, 47, 744, 477...
- **Sai:** 5, 17, 467...

Định nghĩa 2: Số Gần May Mắn (Nearly Lucky Number)

Là số nguyên n bất kỳ, sao cho **số lượng** các chữ số may mắn có trong n phải là một **Số May Mắn**.

Lộ trình tư duy (Chunks)

Chúng ta chia bài toán thành 2 mảnh ghép:

① **Chunk 1: Sàng lọc và Đếm.**

- Tìm xem trong n có bao nhiêu chữ số 4 và 7.

② **Chunk 2: Thăm định kết quả.**

- Kiểm tra xem con số vừa đếm được có phải là số may mắn hay không.

Chunk 1: Sàng lọc và Đếm

Logic (Tư duy ẩn dụ)

Tưởng tượng số nguyên n là một **đoàn tàu dài**. Mỗi toa là một chữ số. Bạn là người soát vé chỉ quan tâm đến **"Khách VIP"**:

- Khách VIP: Số **4** và Số **7**.
- Các số khác (0, 1, 2, 3, 5...): Lờ đi.

Hành động: Cầm một cái bấm đếm số. Cứ gặp 1 khách VIP \rightarrow Bấm "Tạch" (cộng thêm 1).

Cảnh báo quan trọng!

Bẫy (Trap)

Đề bài cho n có thể lên tới 10^{18} (18 chữ số).

- Dùng toán học (chia lấy dư, chia nguyên) vẫn được nhưng phức tạp.
- **Tư duy đơn giản hơn:** Đoàn tàu bản chất là một **Chuỗi ký tự (String)**.

Việc duyệt qua một chuỗi ký tự dễ hình dung hơn nhiều so với xử lý số nguyên lớn.

Thử thách tư duy (Mental Check)

Giả sử đoàn tàu có nội dung: $n = 10047704$

Câu hỏi:

- 1 Bạn tìm thấy những "khách VIP" nào?
- 2 Kết thúc chuyến tàu, biến đếm hiển thị số mấy?

Thử thách tư duy (Mental Check)

Giả sử đoàn tàu có nội dung: $n = 10047704$

Câu hỏi:

- 1 Bạn tìm thấy những "khách VIP" nào?
- 2 Kết thúc chuyến tàu, biến đếm hiển thị số mấy?

Phân tích sai lầm thường gặp

- 1 **Nhầm lẫn "Khách VIP":** Số 1 không phải là số may mắn. Chỉ đếm 4 và 7.
- 2 **Nhầm lẫn ĐẾM và CỘNG:** Chúng ta đếm **số lượng** (tăng 1 đơn vị), không cộng giá trị của các con số lại.

Làm lại Chunk 1 (Sửa sai)

Xét lại: $n = 10047704$

- ❶ $1 \rightarrow$ Bỏ qua.
- ❷ $0 \rightarrow$ Bỏ qua.
- ❸ $0 \rightarrow$ Bỏ qua.
- ❹ $4 \rightarrow \text{VIP!} \rightarrow$ Bấm đếm: 1.
- ❺ $7 \rightarrow \text{VIP!} \rightarrow$ Bấm đếm: 2.
- ❻ $7 \rightarrow \text{VIP!} \rightarrow$ Bấm đếm: 3.
- ❼ $0 \rightarrow$ Bỏ qua.
- ❽ $4 \rightarrow \text{VIP!} \rightarrow$ Bấm đếm: 4.

Kết quả cuối cùng trên máy đếm là: 4.

Thử thách lại (Re-Test)

Để chắc chắn bạn đã nắm được việc **ĐẾM**, hãy xem xét các trường hợp sau:

Trường hợp 1: Input $n = 47$

- Số lượng chữ số may mắn là bao nhiêu?

Thử thách lại (Re-Test)

Để chắc chắn bạn đã nắm được việc **ĐẾM**, hãy xem xét các trường hợp sau:

Trường hợp 1: Input $n = 47$

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 2.** (Số 2 không phải là số may mắn \rightarrow **NO**).

Thử thách lại (Re-Test)

Để chắc chắn bạn đã nắm được việc **ĐẾM**, hãy xem xét các trường hợp sau:

Trường hợp 1: Input $n = 47$

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 2.** (Số 2 không phải là số may mắn \rightarrow **NO**).

Trường hợp 2: Input n toàn số 7 (10 chữ số 7)

- Số lượng chữ số may mắn là bao nhiêu?

Thử thách lại (Re-Test)

Để chắc chắn bạn đã nắm được việc **ĐẾM**, hãy xem xét các trường hợp sau:

Trường hợp 1: Input $n = 47$

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 2.** (Số 2 không phải là số may mắn \rightarrow **NO**).

Trường hợp 2: Input n toàn số 7 (10 chữ số 7)

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 10.** (Số 10 chứa số 1 và 0 \rightarrow **NO**).

Thử thách lại (Re-Test)

Để chắc chắn bạn đã nắm được việc **ĐẾM**, hãy xem xét các trường hợp sau:

Trường hợp 1: Input $n = 47$

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 2.** (Số 2 không phải là số may mắn \rightarrow **NO**).

Trường hợp 2: Input n toàn số 7 (10 chữ số 7)

- Số lượng chữ số may mắn là bao nhiêu?
- **Đáp án: 10.** (Số 10 chứa số 1 và 0 \rightarrow **NO**).

Logic đã hoàn thiện: Chỉ khi số lượng là **4** hoặc **7** thì mới in **YES**.

Chunk 3: Tổng kết & Mã giả (Blueprint)

Chúng ta xử lý n dưới dạng **Chuỗi ký tự (String)**.

Kịch bản chương trình

- ➊ **Nhập liệu:** Đọc cả dòng vào biến chuỗi s .
- ➋ **Chuẩn bị:** Tạo biến đếm $dem = 0$.
- ➌ **Duyệt (Loop):**

```
1 FOR m i ky_tu trong s:  
2     IF ky_tu == '4' OR ky_tu == '7':  
3         dem = dem + 1
```

- ➍ **Phán quyết:**

```
1 IF dem == 4 OR dem == 7:  
2     PRINT "YES"  
3 ELSE:  
4     PRINT "NO"
```

Bước cuối: Hiện thực hóa

Nhiệm vụ của bạn

Hãy viết code hoàn chỉnh dựa trên kịch bản trên bằng ngôn ngữ sở trường (C++, Python, Java...).

Lưu ý nhỏ

- Input là chuỗi (String), không phải số nguyên.
- In hoa **YES** và **NO** đúng yêu cầu.

Chúc các bạn thành công!