ผลบวกผลคูณ (sumproduct)

กลางดึกคืนหนึ่งของค่ายคอมพิวเตอร์โอลิมปิก ไอซี่เดินเข้าห้องเรียนและพบว่าบนกระดานดำมีนิพจน์ (expression) $A[0] + A[1] + \cdots + A[N-1]$ เขียนไว้บนกระดาน โดยที่ $A[0], A[1], \ldots, A[N-1]$ เป็นจำนวนเต็มบวก เพื่อทำให้นิพจน์นี้น่าสนใจขึ้น ไอซี่จึงทำการเปลี่ยนเครื่องหมายบวก $+$ บางอันให้เป็นเครื่องหมายคูณ $\times$ เนื่องจากมีเครื่องหมายบวกอยู่ $N-1$ อัน จึงมีวิธีการเปลี่ยนเครื่องหมายทั้งหมด 2^{N-1} วิธี ไอซี่จึงสงสัยว่าถ้าเกิดพิจารณาทุกวิธีการเปลี่ยนเครื่องหมายที่เป็นไปได้ แล้วผลบวกของค่าของนิพจน์ทั้งหมดที่เป็นไปได้จะมีค่าเท่าไร เขาจึงมาถามคุณ ให้คุณเขียนโปรแกรมเพื่อหาค่าดังกล่าว โดยให้หาเศษจากการหารด้วย $10^9 + 7$

หมายเหตุ. ตามข้อตกลงสากล เราจะทำการคูณก่อนการบวก เช่น $2 + 3 \times 5 = 2 + (3 \times 5)$

รายละเอียดการเขียนโปรแกรม

คุณต้องการเขียนฟังก์ชันต่อไปนี้

```
int sum_product(int N, vector<int> A)
```

โดยให้คืนค่าเศษจากการหารผลบวกทุก 2^{N-1} ค่าเมื่อหารด้วย $10^9 + 7$

เงื่อนไข

- $2 \leq N \leq 100\,000$
- $1 \leq A[i] \leq 10^9$ สำหรับทุกจำนวนเต็ม $0 \leq i < N$

ปัญหาย่อย

1. (10 คะแนน) $N \leq 20$
2. (24 คะแนน) $A[i] = 1$ ทุกจำนวนเต็ม $0 \leq i < N$
3. (22 คะแนน) $N \leq 100$
4. (19 คะแนน) $N \leq 2\,000$
5. (25 คะแนน) ไม่มีเงื่อนไขเพิ่มเติม

ตัวอย่าง

```
sum_product(3, [2, 3, 5])
```

ในตัวอย่างนี้ $N = 3$ และนิพจน์เริ่มต้นคือ $2 + 3 + 5$ จะมีวิธีการเปลี่ยนเครื่องหมายทั้งหมด $2^{3-1} = 4$ วิธี ดังนี้

- $2 + 3 + 5 = 10$
- $2 + 3 \times 5 = 17$
- $2 \times 3 + 5 = 11$
- $2 \times 3 \times 5 = 30$

นั่นคือ คำตอบที่ไอซี่ต้องการคือ $10 + 17 + 11 + 30 = 68$

เกรดเดอร์ตัวอย่าง

เกรดเดอร์ตัวอย่างจะอ่านข้อมูลในรูปแบบดังนี้:

- บรรทัด 1: N
- บรรทัด 2: $A[0]$ $A[1]$ $A[2]$ $\ldots$ $A[N-1]$

เกรดเดอร์ตัวอย่างจะพิมพ์ค่าที่คืนจาก `sum_product`

ขีดจำกัด

- Time limit: 1.0 second
- Memory limit: 512 MB