

Day 1: Practice

A. Binary Search Queries

4 s., 256 MB

กำหนด array a ประกอบด้วยจำนวนเต็ม n ตัวเป็นสมาชิก และจำนวนชุดทดสอบ q ชุด

ชุดทดสอบที่ i คือจำนวน x_i ซึ่งคำตอบของชุดทดสอบชุดที่ i คือ "YES" ถ้าจำนวน x_i ปรากฏใน array a (นั่นคือคำตอบจะเป็น "YES" ถ้ามีอย่างน้อย 1 ตัวใน a เป็น x_i) และคำตอบคือ "NO" สำหรับกรณีอื่น ๆ

ให้หาคำตอบของทุกชุดคำตอบ q ชุดนี้ให้ถูกต้อง

ให้ใช้ **binary search** algorithm ในการแก้โจทย์ข้อนี้ เราจะ **Reject** หากพบว่าใช้ algorithm อื่นใดนอกเหนือจากที่กำหนด

Input

บรรทัดแรกของ input มีจำนวนเต็มสองจำนวนคือ n และ q ($1 \leq n, q \leq 10^5$) — ความยาวของ array a และจำนวนของชุดทดสอบตามลำดับ

บรรทัดที่สองของ input มีจำนวนเต็ม n จำนวน คือ a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) โดยที่ a_i คือสมาชิกลำดับที่ i ของ a

และบรรทัดต่อไป q บรรทัดเป็นจำนวนที่ต้องการทดสอบ ซึ่งในบรรทัดที่ i ประกอบด้วยจำนวนเต็มเพียงจำนวนเดียว x_i ($1 \leq x_i \leq 10^9$).

Output

Print q บรรทัด โดยที่ในบรรทัดที่ i ให้ print "YES" ถ้า x_i เป็นสมาชิกของ a และ "NO" สำหรับกรณีอื่น ๆ

input
6 6 1 2 10 7 2 4 2 8 1000000000 1 4 1
output
YES NO NO YES YES YES

B. Leftmost and Rightmost Occurrences

5 s., 256 MB

กำหนด array a ประกอบด้วยสมาชิกเป็นจำนวนเต็ม n ตัว จำนวนชุดทดสอบ q ชุด และในชุดทดสอบที่ i เป็นจำนวน x_i

คำตอบของชุดทดสอบที่ i คือจำนวนเต็มสองจำนวน l_i และ r_i ซึ่ง l_i เป็นตำแหน่งของจำนวนแรกใน a ที่ x_i ปรากฏ กล่าวคือเป็นตำแหน่งซ้ายที่สุดที่ทำให้ $a[l_i] = x_i$ (หากไม่ปรากฏคำตอบจะเป็น -1 แทน) ในทำนองเดียวกันกับ r_i ซึ่งคือตำแหน่งขวาสุดของ a ที่ปรากฏ x_i นั่นคือ $a[r_i] = x_i$ (หากไม่ปรากฏคำตอบจะเป็น -1 แทน)

ให้ print คำตอบของแต่ละชุดคำตอบ q ชุดนี้

สังเกตว่า l_i สามารถเท่ากับ r_i ได้หากจำนวนที่เราต้องการหานั้นปรากฏเพียงครั้งเดียวใน a

ให้ใช้ **binary search** algorithm ในการแก้โจทย์ข้อนี้ เราจะ **Reject** หากพบว่าใช้ algorithm อื่นใดนอกเหนือจากที่กำหนด

Input

บรรทัดแรกประกอบด้วยจำนวนเต็มสองจำนวน n และ q ($1 \leq n, q \leq 10^5$) — เป็นความยาวของ a และจำนวนชุดทดสอบ q ตามลำดับ

บรรทัดที่สองประกอบด้วยจำนวนเต็ม n จำนวน คือ a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) ซึ่ง a_i คือ สมาชิกตัวที่ i ของ a

บรรทัดต่อไป q บรรทัด สำหรับบรรทัดที่ i ประกอบด้วยจำนวนเต็มหนึ่งตัว x_i ($1 \leq x_i \leq 10^9$)

Output

Print q บรรทัด ซึ่งในบรรทัดที่ i ประกอบด้วยจำนวนเต็มสองจำนวน l_i และ r_i — ตำแหน่งของการปรากฏครั้งแรกซ้ายสุดและขวาสุดของ x_i ใน a (หรือ -1 -1 ถ้า $x_i \notin a$)

input
9 7 2 4 5 2 3 6 4 2 10 2 4 7 10 11 3 2
output
1 8 2 7 -1 -1 9 9 -1 -1 5 5 1 8

C. Nearest Element

5 s., 256 MB

กำหนด array a ประกอบด้วยสมาชิกเป็นจำนวนเต็ม n ตัว โดยมีชุดทดสอบทั้งหมด q ชุด ซึ่งในชุดที่ i มีจำนวนหนึ่งจำนวนคือ x_i

คำตอบของชุดทดสอบที่ i คือสมาชิก a_j ใน a ที่ทำให้ผลต่างสัมบูรณ์ (Absolute difference) ระหว่าง x_i และ a_j มีค่าน้อยที่สุด (กล่าวคือ a_j เป็นสมาชิกของ a ที่ใกล้กับ x_i ที่สุดบนเส้นจำนวน) หากมีสมาชิกหลายค่าที่ทำให้ประโยคดังกล่าวเป็นจริง ให้เลือกจำนวนที่เล็กกว่า

ยกตัวอย่างเช่น ถ้า $a = [1, 7, 5, 3, 2]$ และจำนวนที่ใกล้ 5 ที่สุดคือ $a_3 = 5$ ในขณะจำนวนที่ใกล้กับ 6 ที่สุดคือ $a_3 = 5$ และจำนวนที่ใกล้กับ 4 ที่สุดคือ $a_4 = 3$ เป็นต้น

ให้หาจำนวนที่ใกล้กับ x_i ที่สุดใน a สำหรับชุดทดสอบ q ชุดนี้

ให้ใช้ **binary search** algorithm ในการแก้โจทย์ข้อนี้ เราจะ **Reject** หากพบว่าใช้ algorithm อื่นใดนอกเหนือจากที่กำหนด

Input

บรรทัดแรก ประกอบด้วยจำนวนเต็มสองจำนวน n และ q ($1 \leq n, q \leq 10^5$) — ความยาวของ a และจำนวนชุดทดสอบ ตามลำดับ

บรรทัดที่สองประกอบด้วยจำนวนเต็ม n จำนวน a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) โดยที่ a_i คือสมาชิกตัวที่ i ของ a

q บรรทัดต่อไปคือชุดทดสอบ ซึ่งในบรรทัดที่ i ประกอบด้วยตัวเลขหนึ่งจำนวน x_i ($1 \leq x_i \leq 10^9$)

Output

Print q บรรทัด โดยที่ในบรรทัดที่ i print จำนวนเต็มหนึ่งจำนวนคือ a_j — มีคุณสมบัติคือ $a_j \in a$ และเป็นจำนวนที่ทำให้ $|x_i - a_j|$ มีค่าน้อยที่สุด หากมีสมาชิกหลายค่าที่ทำให้ประโยคดังกล่าวเป็นจริง ให้เลือกจำนวนที่เล็กกว่า

input
9 7 2 4 5 2 3 6 4 2 10 2 4 7 10 11 3 2

output
2
4
6
10
10
3
2

D. Sum Of Two: ยาก

1 s., 256 MB

กำหนด list ของจำนวนเต็ม ที่แตกต่างกัน n ตัวคือ $a = [a_0, a_1, \dots, a_{n-1}]$ และจำนวนเต็ม s

ให้พิจารณาผลบวกที่เป็นไปได้ทั้งหมดของสองจำนวนใด ๆ ที่แตกต่างกันจาก list ดังกล่าว แล้วหาค่าผลบวกที่มากที่สุดที่น้อยกว่าหรือเท่ากับ s

Input

บรรทัดแรกประกอบด้วยจำนวนเต็มบวก n ($2 \leq n \leq 50000$) — จำนวนสมาชิกใน list

บรรทัดที่สองประกอบด้วยเลข n จำนวน ที่แตกต่างกัน ที่คั่นด้วย space a_0, a_1, \dots, a_{n-1} ($-10^9 \leq a_i \leq 10^9$) — จำนวนใน list

บรรทัดที่สามประกอบด้วยจำนวนเต็ม s ($-2 \cdot 10^9 \leq s \leq 2 \cdot 10^9$)

Output

Print จำนวนเต็มหนึ่งจำนวน — ผลรวมมากที่สุดที่เป็นไปได้ระหว่างสองสมาชิกที่ต่างกันใน a ซึ่งน้อยกว่าหรือเท่ากับ s ยืนยันได้ว่าผลรวมดังกล่าวมีค่าตอบแน่นอน

input
5
5 8 9 1 4
15
output
14

input
2
-10 -100
0
output
-110

input
2
1 2
4
output
3

E. But Cardano's Formula Is So Hard :(

1 s., 256 MB

กำหนดฟังก์ชัน $f(x) = 42x^3 + 13x^2 + 1337x$ และจำนวนเต็ม C

ให้หาจำนวนเต็ม x ที่ทำให้ $f(x) = C$

ยืนยันได้ว่ามีคำตอบ x อยู่ และ $0 \leq x \leq 2 \cdot 10^5$

Input

บรรทัดเดียวประกอบด้วย จำนวนเต็มเพียงจำนวนเดียว C ($0 \leq C \leq 10^{18}$)

Output

Print จำนวนเต็ม x เพียงจำนวนเดียวที่ทำให้ $f(x) = C$

input
1392

output
1

input
42263700
output
100

input
0
output
0

F. Fence Building

3 s., 256 MB

เวอยากสร้างรั้วรอบบ้าน (ตอนนี้ยังไม่มียาน) โดยมีไม้กระดาน n แผ่น และความสูงของแผ่นที่ i คือ a_i

เวตต์สินใจ (จริง ๆ มีคนตั้งใจทยอยตัดสินใจไขข้อตัวเอง) ว่าอยากให้รั้วนั้นประกอบไปด้วย อย่างน้อย k แผ่นที่มีความสูงเท่ากับ

วิธีการของเวคือ เขาตัดสินใจตัดแผ่นไม้ให้เป็นสองแผ่น (แต่ใช้กาวติดเชื่อมไม่ได้นะครับ อย่า ๆ ๆ ๆ) และต้องการให้รั้วที่สร้างขึ้นจากอย่างน้อย k แผ่นนี้สูงที่สุดเท่าที่จะเป็นไปได้

ให้หาความสูงที่มากที่สุดที่เป็นไปได้ของรั้วที่สร้างได้ ในบางกรณีแล้วคำตอบอาจจะเป็น 0 ก็ได้

ยกตัวอย่างเช่น ให้ $a = [5, 2, 3, 7, 9]$ และ $k = 4$ จะได้ว่ารั้วที่สร้างได้สูงที่สุดคือ 4 จากการสร้างด้วยวิธีดังต่อไปนี้: เวตัดแผ่นไม้แผ่นแรกให้มีความสูง 4 (จะเหลือ 1 ซึ่งไม่ใช่) แผ่นที่สองกับสามก็ไม่สามารถตัดให้สูง 4 ได้ จึงไม่ใช่ แผ่นที่สี่ตัดให้สูง 4 (เหลือ 3 ไม่ใช่) และแผ่นสุดท้ายตัดให้สูง 4 ได้สองชิ้น แล้วเหลือ 1 ที่ไม่ใช่ จึงได้โดยรวมว่ามีแผ่นไม้ความสูง 4 ที่มีจำนวน $4 = k = 4$ แผ่นพอดี และความสูง 4 นี้แหละที่สูงที่สุดที่จะเป็นไปได้แล้ว

Input

บรรทัดแรกประกอบด้วยจำนวนเต็มสองจำนวน n และ k ($1 \leq n, k \leq 10^5$) — จำนวนของแผ่นไม้ที่เรามี และจำนวนแผ่นไม้ที่น้อยที่สุดในการสร้างรั้วตามลำดับ

บรรทัดที่สองประกอบด้วยจำนวนเต็ม n จำนวน a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) โดยที่ a_i เป็นความสูงของไม้แผ่นที่ i

Output

Print จำนวนเต็มจำนวนเดียว — ความสูงของรั้วที่มากที่สุดที่เวจะสามารถสร้างได้ (คำตอบอาจจะเป็น 0 ได้เช่นกัน)

input
5 4
5 2 3 7 9
output
4

input
5 4
5 2 3 7 10
output
5

input
7 10
21 17 10 5 12 4 22
output
7

G. Math Homework

เวตต้องการแก้โจทย์คณิตศาสตร์ด้วยตัวเอง โดยต้องแก้สมการ อย่างน้อย n สมการ

ในวันแรก เขาแก้โจทย์ได้ v ข้อ วันที่สองอีก $\lfloor \frac{v}{2} \rfloor$ ข้อ วันที่สามอีก $\lfloor \frac{v}{3} \rfloor$ ข้อ และเขายังคงแก้ต่อไป จนวันที่ k ก็แก้ได้อีก $\lfloor \frac{v}{k} \rfloor$ ข้อ

ความหมายของ $\lfloor \frac{x}{y} \rfloor$ หมายถึง x หารด้วย y แล้วปัดเศษลงให้กลายเป็นจำนวนเต็ม

เห็นได้ชัดเลยว่า เมื่อผ่านไปเรื่อย ๆ สักรวันหนึ่ง เขาก็จะแก้สมการได้ 0 ข้อ และหลังจากวันนั้นก็เลยเลิกตั้งใจแก้สมการไปเลย

เขายังไม่ได้ตัดสินใจว่าจำนวนของสมการที่จะแก้ในวันแรก (v) จะเป็นเท่าไร แต่ก็อยากให้ v มีค่าน้อยที่สุดเท่าที่จะเป็นไปได้ และมันจะต้องเพียงพอสำหรับการแก้ อย่างน้อย n สมการเช่นกัน

เวเหนือจะคิดแล้ว เลยมายขอให้น้อง ๆ ช่วยหาคำนวนเต็มทีน้อยที่สุดที่เป็นไปได้ของ v ที่จะเพียงพอสำหรับการแก้ อย่างน้อย n สมการ แต่เขาก็กลัวน้อง ๆ ไม่เข้าใจ จึงมีตัวอย่างมาให้

สมมติให้ $n = 13$ ถ้าเราเลือกว่าวันแรกจะแก้สมการ $v = 6$ ข้อ ดังนั้นจำนวนสมการที่เวจะแก้ได้เมื่อเวลาผ่านไป 6 วันคือ $6 + \lfloor \frac{6}{2} \rfloor + \lfloor \frac{6}{3} \rfloor + \lfloor \frac{6}{4} \rfloor + \lfloor \frac{6}{5} \rfloor + \lfloor \frac{6}{6} \rfloor = 6 + 3 + 2 + 1 + 1 + 1 = 14$ โดยในวันที่ 7 เป็นต้นไปก็จะมีค่าเป็น 0 แต่ในตอนนั้น เวสามารถแก้สมการได้ 14 ข้อ ซึ่งมากกว่า 13 แล้ว ถือว่าใช้ได้ และในเมื่อเราต้องการให้ v มีค่าน้อยที่สุดด้วย ถ้าลองลด v เป็น 5 จะคำนวณจำนวนข้อที่แก้ได้ทั้งหมดว่า $5 + \lfloor \frac{5}{2} \rfloor + \lfloor \frac{5}{3} \rfloor + \lfloor \frac{5}{4} \rfloor + \lfloor \frac{5}{5} \rfloor = 5 + 2 + 1 + 1 + 1 = 10$ ซึ่งไม่สามารถแก้สมการได้ถึง 13 ข้อตามที่ต้องการ ดังนั้นในกรณีนี้ สำหรับ $n = 13$, $v = 6$ จึงเป็นค่าทีน้อยที่สุดและเป็นคำตอบนั่นเอง

Input

บรรทัดเดียวประกอบด้วยจำนวนเต็มหนึ่งตัว n ($1 \leq n \leq 10^6$) — จำนวนสมการที่เวจะต้องแก้เป็นอย่งต่ำ

Output

Print จำนวนเต็มหนึ่งจำนวน — จำนวนเต็ม v ทีน้อยที่สุดที่ทำให้เวแก้สมการได้อย่างน้อย n สมการ

input
1
output
1

input
13
output
6

input
1000000

output
86764

H.. Travel Quest (ver ยาก)

3 s., 256 MB

จินตนาการว่าน้อง ๆ กำลังเล่นเกมคอมพิวเตอร์ ภารกิจต่อไปคือการเดินทางจากจุดที่ 1 ไปยังจุดที่ n

ในเกมมีรถราง m ขบวน โดยขบวนที่ i จะพาไปได้จากจุดใดก็ได้ในช่วง $[l_i, r_i]$ ไปยังจุดใดก็ได้ในช่วงเดียวกันนี้ แต่เราสามารถใช้รถรางขบวนที่ i ได้หากตัวละครของเรา มีเลเวลอย่างน้อย s_i

เริ่มเกมมาเราจะมีเลเวลเท่ากับ 0 และหน้าที่ของเราคือหาเลเวลที่น้อยที่สุดที่ทำให้ตัวละครเราสามารถเคลื่อนที่จากจุดที่ 1 ไปยังจุดที่ n ได้

มีทางเดียวที่เคลื่อนที่ได้คือนั่งรถรางที่กำหนดเท่านั้น ตัวละครเรายังเลเวลต่ำ บินไม่ได้ เหาะเป็นอาชีวะก็ยังไม่ได้เช่นกัน

Input

บรรทัดแรกประกอบด้วยจำนวนเต็มสองจำนวน n และ m ($2 \leq n \leq 10^5; 1 \leq m \leq 10^5$) — จุดสุดท้ายที่ภารกิจของเราอยู่ที่เราต้องไป

และจำนวนของรถรางที่มี ตามลำดับ

m บรรทัดต่อมา ประกอบด้วยข้อมูลของรถรางแต่ละขบวนในแต่ละบรรทัด โดยบรรทัดที่ i มีจำนวนเต็มสามจำนวนคือ l_i, r_i และ s_i ($1 \leq l_i \leq r_i \leq n; 1 \leq s_i \leq 10^9$) — ตำแหน่งซ้ายสุดและขวาสุดของรถรางที่ i และเลเวลที่ต่ำที่สุดที่จะขึ้นรถรางขบวนที่ i ได้ ตามลำดับ

ยืนยันได้ว่าจุดที่ภารกิจอยู่ n จะสามารถเดินทางไปได้จากรถรางที่มีในทุก ๆ ชุดทดสอบ

Output

Print จำนวนเต็มหนึ่งจำนวน — เลเวลที่ต่ำที่สุดที่ต้องการในการเคลื่อนที่จากจุดที่ 1 ไปยังจุดที่ n

input
5 2 1 3 2 3 5 4
output
4

input
5 3 1 3 2 3 5 4 1 5 10
output
4