Note creation request

Josephus Problem কী? f(n)=2l+1\ \text{ where }\ n=2^m+l Josephus Problem হলো একটা বিখ্যাত Mathematical & Computer Science problem। এটা মূলত **...

PASTED

তোমার স্যার আসলে Josephus-এর দুইটা খুব important জিনিস পড়াইছে: 1. **Classic Formula Method** 2. **Recursive Relation Method** আমি একদম শুরু থে...

PASTED

তোমার স্যার আসলে Josephus-এর দুইটা খুব important জিনিস পড়াইছে: 1. **Classic Formula Method** 2. **Recursive Relation Method** আমি একদম শুরু থে...

PASTED

Based on this, give me a clear, detailed, and understandable note

১. Josephus Problem কী?

এটি একটি বিখ্যাত Circular Elimination Problem। কিছু মানুষ গোল হয়ে বসে, নির্দিষ্ট নিয়মে একজন বাদ যায়, এভাবে চলতে থাকে — **শেষে কে বাঁচবে?** এটাই এই problem-এর প্রশ্ন।

> নামটি এসেছে ইতিহাসবিদ **Flavius Josephus** থেকে — যিনি নাকি এই নিয়মটা বুঝে সঠিক জায়গায় দাঁড়িয়ে শেষে বেঁচে গিয়েছিলেন।

Mathematical notation:

$$J(n, k)$$

n = মোট মানুষ | k = প্রতি কত নম্বরে বাদ দেওয়া হবে

২. Step-by-step Example (n=7, k=2)

৭ জন মানুষ, প্রতি ২য় জন বাদ। দেখো কীভাবে elimination হয়:

শুরু

1 2 3 4 5 6 7

↓ 2 বাদ → 4 বাদ → 6 বাদ → 1 বাদ → 5 বাদ → 3 বাদ

শেষে একজন বাকি

~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ ~~6~~ 7

🏆 J(7, 2) = 7 → ৭ নম্বর ব্যক্তি বাঁচে!

৩. Classic Formula (k=2 এর জন্য) সবচেয়ে Important

যখন k = 2 (প্রতি ২য় জন বাদ), তখন একটি shortcut formula আছে।

Step 1: n কে ভাঙো

$$n = 2^m + l$$

n এর চেয়ে ছোট বা সমান সবচেয়ে বড় power of 2 বের করো, বাকিটা হবে l।

Step 2: Formula লাগাও

$$J(n) = 2l + 1$$

Example: n = 7

1. সবচেয়ে বড় power of 2 ≤ 7 → $4 = 2^2$
2. 7 = 4 + 3 → $l = 3$
3. J(7) = 2(3)+1 = 7

Example: n = 10

1. সবচেয়ে বড় power of 2 ≤ 10 → $8 = 2^3$
2. 10 = 8 + 2 → $l = 2$
3. J(10) = 2(2)+1 = 5

Pattern Table

n	Largest 2^m	l	J(n) = 2l+1	Survivor
1	1	0	1	1
2	2	0	1	1
3	2	1	3	3
4	4	0	1	1
5	4	1	3	3
6	4	2	5	5
7	4	3	**7**	**7**

> যখন n নিজেই power of 2 (যেমন 2, 4, 8, 16…) তখন l = 0, তাই সবসময় **J(n) = 1**।

8. Recursive Formula

Programming-এ বেশি ব্যবহার

n যদি Even হয়

$$J(2n) = 2 \cdot J(n) - 1$$

n যদি Odd হয়

$$J(2n+1) = 2 \cdot J(n) + 1$$

> Base case: **J(1) = 1** — শুধু ১ জন থাকলে সে-ই বাঁচে।

Example: J(30) বের করো

```
30 even → J(30) = 2·J(15) − 1
15 odd → J(15) = 2·J(7) + 1
7 odd → J(7) = 2·J(3) + 1
3 odd → J(3) = 2·J(1) + 1 = 2(1)+1 = 3
J(7) = 2(3)+1 = 7
J(15) = 2(7)+1 = 15
J(30) = 2(15)−1 = 29
```

∴ J(30) = 29

General Recursive (যেকোনো k এর জন্য)

$$J(n, k) = (J(n-1, k) + k) \bmod n$$

Base: J(1, k) = 0

৫. Binary Trick 🔥 সবচেয়ে দ্রুত

k=2 এর ক্ষেত্রে সবচেয়ে সহজ trick:

```
সংখ্যাটাকে binary তে লেখো
সবচেয়ে বাম দিকের 1 টা কেটে একদম ডানে লাগাও
ফলাফলকে decimal এ convert করো — ওটাই survivor!
```

n = 10

10 → `1010`

প্রথম 1 শেষে → `0101`

n = 7

7 → `111`

প্রথম 1 শেষে → `111`

= 5 ✓

= 7 ✓

৬. Quick Summary — সব একসাথে

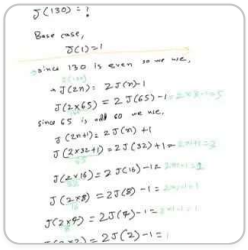
Method	Formula	কখন ব্যবহার
Classic Formula	J(n) = 2l+1, যেখানে $n=2^m+l$	k=2, হাতে দ্রুত calculate
Recursive (even/odd)	J(2n)=2J(n)−1, J(2n+1)=2J(n)+1	k=2, step-by-step
General Recursive	J(n,k)=(J(n-1,k)+k) mod n	যেকোনো k, programming
Binary Trick	1st bit কে শেষে নিয়ে যাও	k=2, সবচেয়ে দ্রুত!
n = power of 2	J(n) = 1 সবসময়	special case

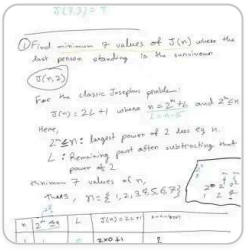
এই note-এ তোমার স্যারের পুরো Josephus Problem টপিক cover করা হয়েছে — problem-এর ধারণা থেকে শুরু করে সব formula, step-by-step example, recursive calculation (J(30) সহ), এবং binary trick।

পরীক্ষার জন্য সবচেয়ে গুরুত্বপূর্ণ যে জিনিসগুলো মাথায় রাখবে:

Classic formula (J(n) = 2l+1) হাতে-কলমে সবচেয়ে দ্রুত, আর **Binary Trick** হলো সবচেয়ে চালাক shortcut — n কে binary তে লিখে প্রথম bit টা শেষে নাও, ব্যস।

কোনো specific অংশ আরো বিস্তারিত বুঝতে চাইলে বলো!





make aslo note from here...and aslo form you..then give e a full note wih before

Check uploaded images >

I can see both images clearly. Let me read them carefully and build the full combined note.

Image 1 shows: J(130) solved via recursive even/odd method, concluding J(130) = 5.
Image 2 shows: CSE 405 class notes — J(n,k) definition, J(9,3) example with circle elimination, and the minimum 7 values table with J(1)=1 through J(7)=7.

Now let me build the complete, comprehensive note:

📘 Course: CSE 405 Josephus + Tower of Hanoi

এই course-এ দুটো classic problem পড়ানো হচ্ছে। এই note টি সম্পূর্ণ Josephus Problem-এর উপর।

১. Josephus Problem — সংজ্ঞা ও ধারণা

Josephus Problem হলো একটি বিখ্যাত **Circular Elimination Problem**। কিছু মানুষ গোল হয়ে দাঁড়ায়, নির্দিষ্ট interval-এ একজন করে বাদ যায়। প্রশ্ন হলো — **শেষে কে বাঁচবে?**

> নামের পেছনে ইতিহাস: ইতিহাসবিদ **Flavius Josephus** নাকি গণনা করে এমন জায়গায় দাঁড়িয়েছিলেন যাতে সে শেষে বেঁচে যায়। এখান থেকেই problem-এর নাম।

Mathematical Notation

$$J(n, k)$$

n	k
Total number of participants (মোট মানুষের সংখ্যা)	Every k-th person will be eliminated (প্রতি k-তম মানুষ বাদ)

> By default, **k = 2** — অর্থাৎ প্রতি ২য় মানুষ বাদ। এটাই সবচেয়ে classic case।

২. Example: J(9, 3) স্যারের class note

৯ জন মানুষ, প্রতি ৩য় জন বাদ যাবে। কে survive করবে?

Elimination Round

শুরু (১ থেকে ৯):

1 2 ~~3~~ 4 5 ~~6~~ 7 8 ~~9~~

Round 1: 3, 6, 9 বাদ

1 2 ~~4~~ 5 7 ~~8~~ 1

Round 2: 4, 8 বাদ (circle continue) → 2 বাদ

1 5 ~~7~~ 1

Round 3: 7 বাদ → 5 বাদ

1

∴ J(9, 3) = 4 → 8 নম্বর ব্যক্তি বাঁচে!

৩. Classic Example: J(7, 2) পূর্ববর্তী note

৭ জন মানুষ, প্রতি ২য় জন বাদ (k=2):

1 ~~2~~ 3 ~~4~~ 5 ~~6~~ 7

বাদ: 2 → 4 → 6 → 1 → 5 → 3

7

∴ J(7, 2) = 7

৪. Classic Formula (k=2 এর জন্য) সবচেয়ে Important

স্যার এই formula দিয়েছেন:

$$J(n) = 2L + 1$$

যেখানে $n = 2^m + L$

$2^m \leq n$ মানে?

n এর চেয়ে ছোট বা সমান **সবচেয়ে বড়** power of 2। একে বলে "largest power of 2 less than or equal to n"।

L মানে?

n থেকে সেই power of 2 বাদ দিলে যা থাকে। $L = n - 2^m$।

Minimum 7 Values of J(n) — স্যারের Table Image 2

n = 1, 2, 3, 4, 5, 6, 7 এর জন্য survivor বের করো:

n	$2^m \leq n$	L	J(n) = 2L+1	Survivor
1	1 (= 2^0)	0	2×0+1	**1**
2	2 (= 2^1)	0	2×0+1	**1**
3	2	1	2×1+1	**3**
4	4 (= 2^2)	0	2×0+1	**1**
5	4	1	2×1+1	**3**
6	4	2	2×2+1	**5**
7	4	3	2×3+1	**7**

স্যারের লেখা result:

J(1)=1, J(2)=1, J(3)=3, J(4)=1
J(5)=3, J(6)=5, J(7)=7

> **Pattern:** যখন n নিজেই 2 এর power (1, 2, 4, 8, 16…), তখন L=0, তাই **J(n) = 1 সবসময়**।

৫. Recursive Formula Step-by-step Method

n যদি **Even** হয় (2n)

$$J(2n) = 2 \cdot J(n) - 1$$

n যদি **Odd** হয় (2n+1)

$$J(2n+1) = 2 \cdot J(n) + 1$$

> **Base case:** J(1) = 1 — শুধু ১ জন থাকলে সে-ই বাঁচে।

Solved Example: J(30) পূর্ববর্তী note

30 (even) → 15 (odd) → 7 (odd) → 3 (odd) → 1 (base)

↓ J(1) = 1 (base case)

↑ J(3) = 2·J(1)+1 = 2(1)+1 = 3

- $J(7) = 2 \cdot J(3)+1 = 2(3)+1 = 7$
- $J(15) = 2 \cdot J(7)+1 = 2(7)+1 = 15$
- $J(30) = 2 \cdot J(15)-1 = 2(15)-1 = 29$

$\therefore J(30) = 29$

Solved Example: J(130) Image 1 — স্যারের note

130 is even → এই chain ধরে নামো:

130 (even) → 65 (odd) → 32 (even) → 16 (even) → 8 (even) → 4 (even) → 2 (even) → 1 (base)

Step 1 — নিচে নামো (breakdown):

1. $J(130) = 2 \cdot J(65) - 1$ [130 even, 130=2×65]
2. $J(65) = 2 \cdot J(32) + 1$ [65 odd, 65=2×32+1]
3. $J(32) = 2 \cdot J(16) - 1$ [32 even, 32=2×16]
4. $J(16) = 2 \cdot J(8) - 1$ [16 even, 16=2×8]
5. $J(8) = 2 \cdot J(4) - 1$ [8 even, 8=2×4]
6. $J(4) = 2 \cdot J(2) - 1$ [4 even, 4=2×2]
7. $J(2) = 2 \cdot J(1) - 1$ [2 even, 2=2×1]

★ $J(1) = 1$ (base case)

Step 2 — উপরে উঠো (back-substitute):

- $J(2\times1) = 2 \cdot J(1)-1 = 2(1)-1 = 1$
- $J(2\times2) = 2 \cdot J(2)-1 = 2(1)-1 = 1$
- $J(2\times4) = 2 \cdot J(4)-1 = 2(1)-1 = 1$
- $J(2\times8) = 2 \cdot J(8)-1 = 2(1)-1 = 1$
- $J(2\times16) = 2 \cdot J(16)-1 = 2(1)-1 = 1$
- $J(2\times32+1) = 2 \cdot J(32)+1 = 2(1)+1 = 3$ [65 odd]

↑ J(2×65) = 2·J(65)−1 = 2(3)−1 = 5 [130 even]

∴ J(130) = 5 → ১৩০ জনের মধ্যে ৫ নম্বর ব্যক্তি বাঁচে!

> **Note:** J(130) Classic formula দিয়েও check করো → 130 = 128 + 2 = 2^7 + 2, তাই L=2, J(130)=2(2)+1=5 ✓ — দুটো method-এই same answer!

General Recursive (যেকোনো k এর জন্য)

$$J(n, k) = (J(n-1, k) + k) \bmod n$$

Base case: J(1, k) = 0

৬. Binary Trick 🔥 সবচেয়ে দ্রুত (k=2)

1. n কে Binary তে লেখো
2. সবচেয়ে বাম দিকের **1** bit টা কেটে একদম ডানে লাগাও (left-rotate)
3. Result টা Decimal-এ convert করো — ওটাই survivor!

n = 10

10 → `1010`
প্রথম 1 শেষে → `0101` = 5
Check: 10=8+2, L=2, 2(2)+1=5 ✓

n = 130

130 → `10000010`
প্রথম 1 শেষে → `00000101` = 5
J(130)=5 ✓ সব method-এ same!

৭. Complete Summary — সব একসাথে

Method	Formula	কখন ব্যবহার করবে
Classic Formula	J(n) = 2L+1, যেখানে $n=2^m+L$	k=2, হাতে দ্রুত calculate
Recursive (even/odd)	J(2n)=2J(n)−1 & J(2n+1)=2J(n)+1	k=2, step-by-step দেখাতে হলে

General Recursive	J(n,k)=(J(n−1,k)+k) mod n	যেকোনো k, programming-এ
Binary Trick	1st bit কে শেষে নিয়ে যাও	k=2, সবচেয়ে দ্রুত shortcut!
Special Case	J(n) = 1, যদি n = 2^m হয়	n power of 2 হলে সবসময় ১ জন বাঁচে

এই full note-এ সব কিছু একসাথে দেওয়া হয়েছে। কোথা থেকে কী নেওয়া হয়েছে সেটা বলি:

Image 1 থেকে: J(130) এর complete step-by-step recursive solution — 130 → 65 → 32 → 16 → 8 → 4 → 2 → 1 করে নেমে আবার back-substitute করে J(130) = 5 বের করা।

Image 2 থেকে: CSE 405 course info, J(n,k) definition (n = total participants, k = elimination interval, default k=2), J(9,3) = 4 এর circle elimination example, এবং minimum 7 values এর table।

আগের note থেকে: J(7,2) example, classic formula, J(30) calculation, binary trick।

আমার পক্ষ থেকে: Binary trick দিয়ে J(130) verify করা, দুই method-এ same answer পাওয়ার confirmation, এবং সব important results একসাথে summary-তে।

now make your note 1 and 2 together

CSE 405 — Josephus Problem

Complete Note

Topics covered	Problem type	Default case
Josephus + Tower of Hanoi	Circular Elimination	k = 2

১. সংজ্ঞা ও ইতিহাস

কিছু মানুষ গোল হয়ে দাঁড়ায়। নির্দিষ্ট interval-এ একজন করে বাদ যায়। শেষে কে বাঁচবে? — এটাই Josephus Problem।

> নামটি এসেছে ইতিহাসবিদ **Flavius Josephus** থেকে। তিনি গণনা করে এমন জায়গায় দাঁড়িয়েছিলেন যাতে সবশেষে বেঁচে যান।

J(n, k)

n	k
Total participants — মোট মানুষ	Every k-th eliminated — প্রতি k-তম বাদ

> By default **k = 2** — প্রতি ২য় জন বাদ যায়। এটাই classic Josephus।

২. Worked Examples — Elimination দেখো

J(7, 2) — ৭ জন, প্রতি ২য় জন বাদ

পূর্ববর্তী note

শুরু:

1 ~~2~~ 3 ~~4~~ 5 ~~6~~ 7

বাদ: 2 → 4 → 6 → (circle) → 1 → 5 → 3

7

J(7, 2) = 7

J(5, 2) — ৫ জন, প্রতি ২য় জন বাদ

1 ~~2~~ 3 ~~4~~ 5

বাদ: 2 → 4 → (circle) → 1 → 5

3

J(5, 2) = 3

J(9, 3) — ৯ জন, প্রতি ৩য় জন বাদ Image 2 — class note

Round 1 — 3, 6, 9 বাদ:

1 2 ~~3~~ 4 5 ~~6~~ 7 8 ~~9~~

Round 2 — circle continue → 4, 8, 2 বাদ:

1 ~~2~~ 5 7 ~~4~~ ~~8~~

Round 3 — 7, 5 বাদ:

4

J(9, 3) = 4 → 8 নম্বর ব্যক্তি বাঁচে

৩. Classic Formula (k = 2) সবচেয়ে Important

$J(n) = 2L + 1$

যেখানে $n = 2^m + L$

$2^m \leq n$

n এর চেয়ে ছোট বা সমান সবচেয়ে বড় power of 2।

L

$L = n - 2^m$ (বাকি অংশ)

Minimum 7 values of J(n) Image 2 — স্যারের table

n	$2^m \leq n$	L	J(n) = 2L+1	Survivor
1	$1 = 2^0$	0	2×0+1	**1**

2	$2 = 2^1$	0	2×0+1	**1**
3	2	1	2×1+1	**3**
4	$4 = 2^2$	0	2×0+1	**1**
5	4	1	2×1+1	**3**
6	4	2	2×2+1	**5**
7	4	3	2×3+1	**7**

স্যারের result: **J(1)=1, J(2)=1, J(3)=3, J(4)=1, J(5)=3, J(6)=5, J(7)=7**

> **Special case:** n নিজেই power of 2 হলে (1, 2, 4, 8, 16…) → L=0 → **J(n) = 1 সবসময়**

Formula দিয়ে quick solve

n = 10

1. $10 \leqslant 8 = 2^3$ → largest power = 8
2. L = 10 − 8 = 2
3. J(10) = 2(2)+1 = 5

n = 130

1. largest power ⩽ 130 → $128 = 2^7$
2. L = 130 − 128 = 2
3. J(130) = 2(2)+1 = 5

8. Recursive Formula Step-by-step Method

n **Even** হলে

$$J(2n) = 2 \cdot J(n) - 1$$

n **Odd** হলে

$$J(2n+1) = 2 \cdot J(n) + 1$$

> **Base case:** J(1) = 1

Solved: J(30) পূর্ববর্তী note

30 even → 15 odd → 7 odd → 3 odd → J(1)=1

- ↑ J(3) = 2·J(1)+1 = 2(1)+1 = 3
- ↑ J(7) = 2·J(3)+1 = 2(3)+1 = 7
- ↑ J(15) = 2·J(7)+1 = 2(7)+1 = 15

↑ J(30) = 2·J(15)−1 = 2(15)−1 = 29

J(30) = 29

Solved: J(130) Image 1 — স্যারের note

130 even → 65 odd → 32 even → 16 even → 8 even → 4 even → 2 even → J(1)=1

নিচে নামো (breakdown):

1. J(130) = 2·J(65) − 1 [130 = 2×65, even]
2. J(65) = 2·J(32) + 1 [65 = 2×32+1, odd]
3. J(32) = 2·J(16) − 1 [32 = 2×16, even]
4. J(16) = 2·J(8) − 1 [16 = 2×8, even]
5. J(8) = 2·J(4) − 1 [8 = 2×4, even]
6. J(4) = 2·J(2) − 1 [4 = 2×2, even]
7. J(2) = 2·J(1) − 1 [2 = 2×1, even]

★ J(1) = 1 (base case)

উপরে উঠো (back-substitute):

↑ J(2) = 2(1)−1 = 1

↑ J(4) = 2(1)−1 = 1

↑ J(8) = 2(1)−1 = 1

↑ J(16) = 2(1)−1 = 1

↑ J(32) = 2(1)−1 = 1

↑ J(65) = 2·J(32)+1 = 2(1)+1 = 3 [odd]

↑ J(130) = 2·J(65)−1 = 2(3)−1 = 5 [even]

J(130) = 5 → ১৩০ জনের মধ্যে ৫ নম্বর ব্যক্তি বাঁচে!

> Cross-check: Classic formula দিয়েও → 130 = 128+2 = 2^7+2, L=2, J(130) = 2(2)+1 = **5** ✓ — দুটো method-এ same!

General Recursive — যেকোনো k এর জন্য

$$J(n, k) = (J(n-1, k) + k) \bmod n$$

Base case: $J(1, k) = 0$

৫. Binary Trick সবচেয়ে দ্রুত — k=2

1. n কে Binary তে লেখো
2. সবচেয়ে বাম দিকের **1** bit টা কেটে একদম ডানে লাগাও (left rotate)
3. Result টা Decimal-এ convert করো — ওটাই survivor!

n = 7
7 → `111`
rotate → `111` = 7 ✓

n = 10
10 → `1010`
rotate → `0101` = 5 ✓

n = 130
130 → `10000010`
rotate → `00000101` = 5 ✓

৬. CS Applications

- Circular Linked List
- Scheduling Algorithm
- Competitive Programming
- Recursion
- Data Structures
- Game Theory

৭. Master Summary — সব একসাথে

Method	Formula	কখন ব্যবহার
Classic Formula	$J(n)=2L+1,\ n=2^m+L$	k=2, হাতে দ্রুত
Recursive even/odd	$J(2n)=2J(n)-1$ & $J(2n+1)=2J(n)+1$	k=2, step-by-step
General Recursive	$J(n,k)=(J(n-1,k)+k) \bmod n$	যেকোনো k
Binary Trick	1st bit শেষে নাও → decimal	k=2, fastest!

n = power of 2	J(n) = 1	Special case

Key Results মনে রাখো

এটা হলো তোমার তিনটা note একসাথে — একটা complete, unified Josephus reference।

ঠিক কী কী merge হয়েছে:

পূর্ববর্তী Note 1 (text documents) থেকে — J(7,2), J(5,2) এর step-by-step elimination, classic formula ও তার table, J(30) recursive solve, binary trick।

Images (Note 2 — স্যারের class) থেকে — CSE 405 course info, J(n,k) definition with n ও k এর meaning, default k=2, J(9,3)=4 circle elimination, minimum 7 values এর table, এবং J(130) এর পুরো recursive breakdown।

আমার addition — তিনটা example-এই cross-check (classic formula ও recursive দুটো দিয়েই same answer confirm), binary trick দিয়ে J(130) verify, এবং সব key results একসাথে summary-তে।

Classic Formula J(n)=2L+1, n=2^m+L k=2, হাতে দ্রুত ..assa k jodi 2 na hoi takon ki ai formula kaj kore na?

হ্যাঁ, একদম সঠিক ধরেছ।

Classic Formula `J(n) = 2L+1` শুধুমাত্র **k=2** এর জন্যই কাজ করে। k অন্য কিছু হলে এই formula আর valid না।

কারণটা হলো — এই formula টা derive করা হয়েছে specifically প্রতি ২য় জন বাদ দেওয়ার pattern ধরে। সেই pattern-এ একটা সুন্দর binary structure আছে বলেই shortcut সম্ভব হয়। k=3 বা k=5 হলে সেই pattern ভেঙে যায়, আর অত সহজ closed-form formula পাওয়া যায় না।

তাই k≠2 হলে তোমাকে **General Recursive Formula** ব্যবহার করতে হবে:

J(n, k) = (J(n−1, k) + k) mod n

Base case: J(1, k) = 0

এটাই যেকোনো k এর জন্য কাজ করে। যেমন J(9, 3) তুমি দেখেছ — সেটা এই general formula দিয়েই solve হয়, classic formula দিয়ে নয়।

সংক্ষেপে বললে — Classic formula হলো k=2 এর একটা special shortcut, আর general recursive হলো universal solution।

Your last note was awesome, but I cannot copy and paste it in my doc

তুমি চাও note টা একটা downloadable file হিসেবে পাও — Word doc বা PDF হিসেবে?

Q: কোন format-এ চাও?
A: Plain text (.txt)

Ran a command, read a file

এই file-এ পুরো note আছে — সব section, সব formula, সব examples সহ। Download করে যেকোনো জায়গায় copy-paste করতে পারবে।

Josephus note
TXT
Download

tmr html tar moto paoah jabe na?

HTML file বানিয়ে দিচ্ছি — একদম আগের note-এর মতো দেখতে, download করে browser-এ open করলেই সুন্দর render হবে।

Viewed a file, created a file

Check frontend design skill

Full Josephus note as a beautiful standalone HTML file

josephus_note.html

Done

amar formula gula sob milla jillah gece ..awlai gece..