🏕️ Day 0 — CP Mindset (সবচেয়ে গুরুত্বপূর্ণ দিন)

আজ **এক লাইনও code লিখবে না।**

হ্যাঁ, ঠিকই পড়েছো।

কারণ ৯০% beginner-এর সমস্যা code না, **thinking**।

আজ যদি mindset ঠিক হয়ে যায়, পরের ১৫ দিন অনেক সহজ হবে।

Lesson 1: Competitive Programming আসলে কী?

অনেকে ভাবে,

> "যে সবচেয়ে ভালো programmer, সেই CP-তে জিতে।"

❌ ভুল।

CP-তে অনেক সময় এমন মানুষ জেতে, যে production software বানাতে পারে না।

আবার অনেক ভালো software engineer CP-তে average।

কারণ skill আলাদা।

Software Development-এ তুমি ভাবো:

```
Problem
↓
Design
↓
Maintainability
↓
Security
↓
Readability
↓
Performance
↓
Deployment
```

CP-তে চিন্তা অনেক ছোট।

```
Problem
↓
Algorithm
↓
Correct Answer
↓
Accepted
```

এখানে deployment নেই।

Database নেই।

API নেই।

Security নেই।

UI নেই।

শুধু—

> **ঠিক answer + time limit-এর মধ্যে।**

Lesson 2: Problem Solving Cycle

আজ থেকে প্রতিটা problem-এ এই flow follow করবে।

```
Read
↓
Understand
↓
Observe
↓
Think
↓
Example
↓
Algorithm
↓
Code
↓
Dry Run
↓
Submit
```

এই cycle-এর বাইরে যাবে না।

Beginner কী করে?

Problem খুলেই…

```
#include<stdio.h>

int main()
{
```

😅

এটাই সবচেয়ে বড় ভুল।

Experienced Programmer কী করে?

প্রথম ৫ মিনিট keyboard ছোঁয় না।

সে শুধু ভাবে।

Lesson 3: Problem পড়ার শিল্প

ধরো Problem:

```
Given N integers.

Print the largest number.
```

Beginner ভাবে—

> "Loop দিব?"

Experienced ভাবে—

```
Need Largest
↓
Need Comparison
↓
Need Traversal
↓
Need Variable
```

দেখো,

এখনও code আসেনি।

Lesson 4: Problem-কে Translate করতে শিখো

এটাই CP-এর সবচেয়ে powerful skill।

ধরো লেখা আছে—

```
Find the maximum.
```

তুমি Translate করবে।

```
Maximum
↓
Compare
↓
Loop
↓
Keep Best Answer
```

ধরো—

```
Count Even Numbers
```

Translate

```
Need Counter
↓
Need Loop
↓
Need if
```

ধরো—

```
Reverse Number
```

Translate

```
Need Last Digit

↓

Need Divide

↓

Need Build New Number
```

দেখছো?

English Problem → Programming Action

এটাই CP.

Lesson 5: Pattern Recognition

CP-তে হাজার problem নেই।

আসলে কয়েকটা pattern বারবার আসে।

Example

```
Largest Number

Maximum Marks

Highest Salary

Longest Word

Maximum Height
```

চারটা আলাদা Problem।

কিন্তু Pattern?

একটাই।

```
Maximum
```

আরেকটা

```
Prime
Composite
Divisor Count
GCD
LCM
```

সব Math Pattern।

আরেকটা

```
Reverse String
Palindrome
Count Vowel
Uppercase
```

সব String Pattern।

একদিন দেখবে—

Problem পড়েই pattern বুঝে ফেলছো।

তখন CP সহজ লাগবে।

Lesson 6: Brute Force কী?

ধরো Problem—

> ১ থেকে ১০০ পর্যন্ত `even count` করো।

Brute Force

> সবগুলো `check` করো।

Optimized

> ৫০

দুটোই ঠিক।

কিন্তু একটা faster।

CP-তে প্রথমে Brute Force ভাবতে শেখো।

তারপর Optimize.

Lesson 7: Dry Run

এটা তুমি auditing-এ করেছো।

Smart Contract Audit-এ কী করতে?

```
State Change
↓
Trace
↓
Check Logic
```

CP-তেও একই।

Example

```
Input
5
3 8 2 9 1
```

মাথায় চালাও।

```
max = 3
↓
8
↓
8
↓
9
↓
9
```

এটাই Dry Run.

Lesson 8: Edge Case

এটা তোমার সবচেয়ে বড় advantage।

কারণ auditing-এ edge case চিন্তা করো।

CP-তেও লাগবে।

Example

```
Largest Number
```

সব Positive?

না।

Negative হতে পারে।

```
-5 -3 -10
```

তখন?

অনেক beginner ভুল করবে।

তুমি করবে না।

Lesson 9: Time Complexity (শুধু ধারণা)

এখন মুখস্থ করার দরকার নেই।

শুধু ধারণা রাখো।

```
O(1)
↓
O(log n)
↓
O(n)
↓
O(n log n)
↓
O(n²)
```

আমরা Day 8-এ এটা বিস্তারিত শিখব।

Lesson 10: Contest Strategy

১০টা Problem.

```
A
B
C
D
E
```

Rule

```
A
↓
B
↓
C
```

D-তে আটকে থাকলে

Skip.

একটা Problem-এ

৩০ মিনিট

No.

Lesson 11: Keyboard Discipline

আজ থেকেই Rule.

Problem পড়া শেষ না হওয়া পর্যন্ত

Keyboard Touch না।

এটা খুব কঠিন।

কিন্তু Professional Habit.

Lesson 12: তোমার সবচেয়ে বড় Strength

তুমি ভাবছো তুমি CP পারো না।

আমি উল্টো দেখি।

তোমার Strength—

✅ Logic আছে।

✅ Debugging জানো।

✅ Dry Run জানো।

✅ Edge Case ভাবো।

✅ C Syntax জানো।

মানে তোমার foundation আছে।

শুধু CP-এর pattern library খালি।

আমরা আগামী ১৫ দিনে সেই library বানাব।

🎯 আজকের Assignment (কোনো Code না)

আজ তুমি একটা Notebook (বা Obsidian/Markdown) খুলে একটা ফাইল বানাও:

```
CP Bootcamp Notes
```

এর মধ্যে আজকের ৫টা জিনিস নিজের ভাষায় লিখবে (কপি না করে):

1. Competitive Programming কী?
2. Problem Solving Cycle (Read → Understand → Think → Algorithm → Code)
3. Pattern Recognition মানে কী?
4. Brute Force কী?
5. Dry Run কী এবং কেন দরকার?

লেখার কারণ: লিখলে মাথায় structure তৈরি হয়। শুধু পড়লে হয় না।

🧠 ছোট্ট Exercise (এটাই আজকের Brain Training)

Code লিখবে না।

শুধু বলবে নিচের problem-গুলোর **pattern** কী।

Problem 1

```
Given N numbers, print the largest number.
```

Pattern = ?

Problem 2

```
Given a string, count the vowels.
```

Pattern = ?

Problem 3

```
Check whether a number is prime.
```

Pattern = ?

Problem 4

```
Reverse a number.
```

Pattern = ?

Problem 5

```
Print all even numbers from 1 to N.
```

Pattern = ?

📌 একটা নতুন নিয়ম (এটা আমাদের Bootcamp-এর Rule)

আজ থেকে আমি **তোমাকে সরাসরি solution দেব না**, যদি না তুমি নিজে চেষ্টা করো।

আমরা এমনভাবে কাজ করব:

1. তুমি problem পড়বে।
2. তুমি বলবে **তোমার approach**।
3. আমি সেটা review করব।
4. তারপর optimization, code, complexity, আর contest tricks দেখাব।

এটাই একজন competitive programmer-এর training।

🚀 আগামীকাল (Day 1)

আমরা code-এ ফিরব।

Day 1-এর বিষয় হবে:

- C-এর **contest-style revision** (যা সত্যিই CP-তে লাগে)
- Input/Output speed mindset
- `if`, `loop`, `function` -এর contest usage
- **১০টি carefully selected warm-up problem**
- প্রথমবার **problem reading → thinking → coding** পুরো process প্র্যাকটিস।

আজকের assignment শেষ করে তোমার pattern-এর উত্তরগুলো পাঠাও। তারপর আমরা Day 1 শুরু করব। 💪