📘 CP Bootcamp: From Zero to First Contest (Bangla Edition)

Welcome to **Day 1** of our Competitive Programming journey!

> > This lesson is designed to build a strong foundation before solving programming problems. Follow the chapters in order to understand the concepts step by step.

📚 Day 1 - Table of Contents

Chapter 1 — Contest Programmer Mindset

- Programmer vs. Competitive Programmer
- What to Think Before a Contest
- Common Beginner Mistakes

Chapter 2 — Contest Environment Setup

- VS Code Setup
- Project & Folder Structure
- Compile Command
- Run Command
- Fast Development Workflow
- Arch Linux Configuration

Chapter 3 — C for Competitive Programming

Essential Topics

- Input & Output (`scanf` , `printf`)

- Variables & Data Types
- Conditional Statements (`if` , `else`)
- Loops
- Functions
- Arrays
- Strings

Topics We Don't Need Right Now

- (Will be discussed later)

Chapter 4 — Contest Coding Style

- Recommended Code Template
- Why This Template?
- Code Formatting
- Brace Style
- Indentation
- Variable Naming Convention
- Writing Clean & Readable Code

Example:

```
#include <stdio.h>

int main(void)
{
    return 0;
}
```

Chapter 5 — Problem Reading Framework

Learn how to understand a programming problem before writing any code.

```
Problem
   ↓
Input
   ↓
Output
   ↓
Constraints
   ↓
Observe Patterns
   ↓
Examples
   ↓
Brute Force Idea
   ↓
Algorithm
   ↓
Code
```

Chapter 6 — Warm-up Problems

Recommended Practice Order (HackerRank)

1. Sum and Difference of Two Numbers
2. Functions in C
3. Pointers in C
4. Conditional Statements
5. Loops
6. ...

> The complete practice roadmap will be provided in the lesson.

Chapter 7 — Problem Analysis Guide

For every practice problem, we'll discuss:

- Expected Time
- Difficulty Level

- Pattern Recognition
- Why Are We Solving This Problem?
- Key Learning Outcome

Chapter 8 — Homework

- Practice Tasks
- Assignment Submission
- What to Write
- Review Guidelines

Chapter 9 — Revision Sheet

A one-page summary of everything learned today.

- Key Concepts
- Important Syntax
- Common Mistakes
- Contest Tips
- Quick Revision Notes

Day 1 — Chapter 1

The Competitive Programmer's Mindset

> **"Programming is writing code. Competitive Programming is solving problems."**

Welcome Back! 👋

প্রথমেই...

Congratulations!

তুমি **Day 0** শেষ করেছো।

অনেকেই Day 0-কে boring মনে করে skip করে।

কিন্তু আমি ইচ্ছা করেই Day 0-তে **এক লাইনও Code লিখাইনি।**

কেন?

কারণ...

আমি তোমাকে Typist বানাতে চাই না।

আমি তোমাকে Problem Solver বানাতে চাই।

🎯 Chapter Goal

এই Chapter শেষে তুমি বুঝতে পারবে—

- Competitive Programmer কীভাবে চিন্তা করে।
- কেন Code সবসময় প্রথম Step না।
- Problem পড়ার সময় মাথার ভিতরে কী চলা উচিত।
- কেন একজন Beginner আর একজন Expert একই Problem-কে আলাদা চোখে দেখে।

একটা গল্প...

ধরো...

একটা University Contest হচ্ছে।

দুইজন Contestant আছে।

👨‍🎓 Student A

C Language খুব ভালো পারে।

Loop পারে।

Function পারে।

Pointer পারে।

String পারে।

Structure পারে।

File পারে।

সব পারে।

Question পেল।

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

সে কী করল?

```
#include<stdio.h>

int main()
{
```

সে Code লেখা শুরু করে দিল।

👨‍💻 Student B

C Language Student A-এর মতো ভালো না।

কিন্তু...

সে Code লেখে না।

সে Problem-টা পড়ে।

সে কাগজে লিখে—

```
Need

↓

Maximum

↓

Comparison

↓

Traversal

↓

Store Best Answer
```

তারপর Code লেখে।

Contest শেষে কী হলো?

Student B আগে Solve করল।

কেন?

কারণ...

সে Code লেখেনি।

সে Problem Solve করেছে।

⚠ Lesson 1

Programming ≠ Problem Solving

এটা পুরো Bootcamp-এর সবচেয়ে Important Line.

Programming হচ্ছে—

> Computer-কে Instruction দেওয়া।

Competitive Programming হচ্ছে—

> Problem-কে Algorithm-এ Convert করা।

একটা Example দেখি।

Problem:

```
Print the sum of N numbers.
```

Programming Knowledge

```
scanf()

printf()

for()

int
```

Problem Solving Knowledge

```
Need Sum
↓
Need Traversal
↓
Need Accumulator
```

দেখলে?

Programming আর Problem Solving আলাদা।

Mentor Note 💡

যদি কেউ আমাকে জিজ্ঞেস করে—

> "CP-এর সবচেয়ে Important Skill কী?"

আমি কখনো বলব না—

> C Language

আমি বলব—

> **Pattern চিনার ক্ষমতা।**

Lesson 2

Code হচ্ছে Last Step

Beginner-এর Flow

```
Problem

↓

Open VS Code

↓

main()

↓

Thinking
```

😂

Professional-এর Flow

```
Problem
↓
Reading
↓
Thinking
↓
Example
↓
Pattern
↓
Algorithm
↓
Code
```

Code এখানে **শেষে**।

একটা বাস্তব উদাহরণ

ধরো Problem—

```
Input

5

Output

120
```

অনেকে সাথে সাথে লিখবে—

```
int fact=1;
```

না।

Professional কী করবে?

সে প্রথমে জিজ্ঞেস করবে—

```
120 কেন?

5 থেকে 120 হলো কীভাবে?

1×2×3×4×5?
```

এখন সে বুঝল—

```
Problem

↓

Factorial
```

এখন Code লেখা যাবে।

Lesson 3

Read Like a Detective

Contest-এ একটা Problem তিনবার পড়া খুবই Normal.

অনেকে ভাবে—

> "Problem তো পড়েছি।"

আসলে পড়েনি।

সে শুধু চোখ বুলিয়েছে।

আমি চাই তুমি Detective-এর মতো পড়ো।

ধরো লেখা আছে—

```
Given N integers,
find the largest EVEN number.
```

Beginner কী দেখে?

```
Largest Number
```

Professional কী দেখে?

```
Largest

+

Even
```

এই "Even" শব্দটাই পুরো Problem বদলে দিল।

🧠 Think Like This

Problem পড়ার সময় নিজেকে প্রশ্ন করো—

Question 1

Input কী?

```
Number?

Array?

String?

Matrix?
```

Question 2

Output কী?

```
Print?

Return?

Count?

Maximum?

Minimum?
```

Question 3

Keyword কী?

```
Maximum

Minimum

Count

Prime

Palindrome

Reverse

Sort
```

Question 4

কোন Pattern?

```
Math?

Array?

String?

Simulation?
```

Question 5

Brute Force কী হতে পারে?

এই পাঁচটা Question তোমার Brain-এর Default হয়ে যাবে।

Lesson 4

Keyboard Addiction

এটা Beginner-এর সবচেয়ে Dangerous Habit.

Problem পেলেই—

```
#include<stdio.h>

int main()
{
```

Stop.

আজ থেকে Rule.

Keyboard ধরার আগে ৫ মিনিট Problem নিয়ে ভাববে।

Mentor Note 💡

আমি Smart Contract Audit করার সময়ও...

Code পড়ার আগে...

আমি Contract-এর Purpose বুঝি।

CP-তেও একই।

Code-এর আগে Problem.

Lesson 5

Every Problem is a Story

অনেকে Problem দেখে—

```
Boring.
```

Professional দেখে—

```
Story.
```

Example

```
There are N students.

Find the highest marks.
```

Story Ignore.

Actually Problem হচ্ছে—

```
Maximum
```

আরেকটা

```
There are N boxes.

Count empty boxes.
```

Story Ignore.

Problem হচ্ছে—

```
Counting
```

তুমি যদি Story-এর ভিতর থেকে Pattern বের করতে পারো...

তাহলেই CP সহজ হয়ে যাবে।

🏆 Contest Tip #1

Contest-এ Problem-এর English যত বড়...

Solution অনেক সময় তত ছোট।

ভয় পাবে না।

Common Beginner Mistakes ⚠

❌ Mistake 1

Problem একবার পড়ে Code লেখা।

❌ Mistake 2

Sample Input না দেখা।

❌ Mistake 3

Sample Output Ignore করা।

❌ Mistake 4

Keyword না খোঁজা।

❌ Mistake 5

Story-তে আটকে যাওয়া।

Golden Rule ⭐

```
Read
↓
Understand
↓
Think
↓
Translate
↓
Pattern
↓
Algorithm
↓
Code
```

এই Flow পুরো Bootcamp-এ Follow করবে।

Real Developer Insight

তুমি Security নিয়ে কাজ করেছো।

তুমি যখন Smart Contract Audit করো...

তখন কি সাথে সাথে Vulnerability লিখে ফেলো?

না।

তুমি—

```
Contract বুঝো

↓

State বুঝো

↓

Flow বুঝো

↓

Edge Case ভাবো

↓

তারপর Finding লেখো।
```

Competitive Programming-ও ঠিক একই।

```
Problem বুঝো

↓

Pattern বুঝো

↓

Algorithm ভাবো

↓

তারপর Code লেখো।
```

তোমার Audit Experience এখানে তোমাকে Help করবে।

📌 Chapter 1 Summary

আজ আমরা শিখলাম—

- Programming আর Problem Solving এক না।
- Code হচ্ছে শেষ Step।
- Problem Detective-এর মতো পড়তে হয়।
- Story Ignore করে Pattern খুঁজতে হয়।
- Keyboard ধরার আগে Brain ব্যবহার করতে হয়।

📝 Mini Assignment

আজ Code লিখবে না।

নিচের ৫টা Problem-এর শুধু Pattern লিখবে।

Problem 1

```
There are N students.

Print the student with the highest marks.
```

Problem 2

```
Given a string,

count uppercase letters.
```

Problem 3

```
Check whether a number is divisible by both 3 and 5.
```

Problem 4

```
Print all numbers divisible by 7 from 1 to N.
```

Problem 5

```
Reverse an array.
```

🧠 Reflection

আজকের শেষে নিজের কাছে এই তিনটা Question করবে।

১.

আমি কি Problem পড়ে সাথে সাথে Code লিখতে যাই?

২.

আমি কি Story পড়ি নাকি Pattern খুঁজি?

৩.

আমি কি Keyboard-এর আগে Brain ব্যবহার করি?

🎯 Mentor's Final Words

আজকের Chapter-এর সবচেয়ে গুরুত্বপূর্ণ লাইনটা আবার লিখছি—

> **"একজন Beginner Code লেখে। একজন Competitive Programmer Problem Solve করে।"**

আজ থেকে তুমি আর শুধু Code লিখবে না।

তুমি Problem Solve করতে শিখবে। 🔥

Day 1 — Chapter 2

Setting Up Your Battlefield

A Professional Competitive Programmer's Workspace

> **"A soldier never enters the battlefield without preparing his weapon."**

তুমিও Contest-এ যাওয়ার আগে তোমার **Environment** প্রস্তুত করবে।

🎯 Chapter Goal

এই Chapter শেষে তুমি জানতে পারবে—

- একজন Competitive Programmer-এর Workspace কেমন হওয়া উচিত।
- Folder Structure কীভাবে সাজাতে হয়।
- VS Code কীভাবে ব্যবহার করলে সময় বাঁচে।
- Arch Linux-এ Compile & Run-এর Fast Workflow।
- Contest-এর সময় ৫–১৫ সেকেন্ড কীভাবে বাঁচানো যায়।

Lesson 1

Environment Matters

অনেক Beginner ভাবে—

> "Programming মানেই Code."

না।

Professional Programmer-এর জন্য...

Environment-ও Skill।

একটা Example।

Contest শুরু হয়েছে।

দুইজন Programmer।

Student A

Code লিখল।

Compile করতে গেল।

```
gcc main.c
```

Oops.

```
main.c not found
```

সে ভুল Folder-এ আছে।

৩০ সেকেন্ড নষ্ট।

Student B

ঠিক Structure।

Shortcut।

Compile।

Run।

১০ সেকেন্ড।

Contest-এ

৩০ সেকেন্ড

×

২০ বার

=

১০ মিনিট।

Mentor Note 💡

Contest অনেক সময় **Programming Contest না।**

এটা **Time Management Contest**।

Lesson 2

Professional Folder Structure

আমি চাই না...

তোমার Desktop এমন হোক।

```
main.c

main(1).c

new.c

test.c

final.c

final2.c

final_final.c
```

😅

Professional Structure

```
CP-Bootcamp/
│
├── HackerRank/
│
├── CSES/
│
├── Codeforces/
│
├── Contest/
│
├── Notes/
│
├── Templates/
│
└── Practice/
```

এর ভিতরে

Example

```
HackerRank/
│
├── Day01/
│      sum.c
│      loop.c
│      function.c
│
├── Day02/
│
└── Day03/
```

কেন?

কারণ...

দুই মাস পরে

তুমি সহজেই

যেকোনো Code খুঁজে পাবে।

Lesson 3

File Naming Convention

ভুল

```
abc.c

test.c

new.c

main.c
```

Professional

```
01_sum.c

02_even.c

03_factorial.c

04_prime.c
```

আরও ভালো

```
cf_71A.c

cses_weird_algorithm.c

hr_functions.c
```

এক দেখাতেই বুঝবে

Code কোথাকার।

Lesson 4

VS Code Setup

আমরা Fancy Theme শিখব না।

Productivity শিখব।

Extensions

✅ C/C++

Microsoft

✅ Error Lens

Compile Error চোখের সামনে দেখাবে।

✅ Better Comments

✅ Code Runner

না।

কেন Code Runner না?

কারণ...

Contest-এ

Code Runner থাকে না।

Compiler থাকে।

আমরা

Compiler ব্যবহার করব।

Lesson 5

Arch Linux Workflow

তুমি Arch ব্যবহার করো।

আমরা Terminal থেকে Compile করব।

Example

```
gcc main.c -o main
```

Run

```
./main
```

এক লাইনে

```
gcc main.c -o main && ./main
```

Mentor Tip 💡

আজ থেকেই Terminal-এর বন্ধু হও।

Competitive Programmer-এর অর্ধেক শক্তি

Terminal।

Lesson 6

Input File দিয়ে Test করা

Professionalরা কী করে?

বারবার Input লেখে না।

input.txt

```
5

1 2 3 4 5
```

Compile

```
gcc main.c -o main
```

Run

```
./main < input.txt
```

Output

Screen-এ আসবে।

আরও Professional

```
./main < input.txt > output.txt
```

এখন

output.txt

খুলে Result দেখো।

কেন?

কারণ Contest-এর আগে

নিজের Test বানাতে শিখতে হবে।

Lesson 7

Build Your Own Test Cases

Beginner

শুধু Sample Input।

Professional

নিজেই নতুন Test বানায়।

Example

Problem

Largest Number

Sample

```
5

1 4 7 2 3
```

Professional আরও Test করবে।

```
1

10
```

```
5

-5 -2 -8 -1 -9
```

```
5

7 7 7 7 7
```

```
2

1000000 999999
```

এগুলোই

Edge Case।

Lesson 8

Keyboard Shortcuts

VS Code-এ এগুলো মুখস্থ করো।

Shortcut	কাজ
Ctrl + `	Terminal Open
Ctrl + P	File Search
Ctrl + /	Comment
Ctrl + Shift + P	Command Palette
Ctrl + S	Save

Linux Terminal

Command	কাজ
`cd`	Folder Change
`ls`	File List
`mkdir`	Folder Create
`touch`	File Create
`clear`	Screen Clear

Lesson 9

Compile Errors are Friends

অনেক Beginner...

Red Error দেখলেই Panic।

Professional...

Error পড়ে।

Example

```
expected ';'
```

মানে?

Compiler তোমাকে বলছে

Semicolon Missing।

Compiler

Enemy না।

Friend।

Lesson 10

Create a Contest Template

আজ থেকেই

একটা Template বানাও।

```
#include <stdio.h>

int main(void)
{
    return 0;
}
```

এখন প্রশ্ন—

> "Sir, এত ছোট Template কেন?"

কারণ...

আমি Template-এ অপ্রয়োজনীয় Code বিশ্বাস করি না।

অনেকে লিখে—

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<math.h>
```

যখন দরকারই নেই।

Mentor Note 💡

Professional Programmer

কম Code লিখে।

কারণ...

কম Code

=

কম Bug।

Lesson 11

The 5-Second Rule

Contest শুরু।

Problem Open।

Keyboard ধরবে?

না।

এই ৫ সেকেন্ড

শুধু ভাববে।

```
Input?
↓
Output?
↓
Pattern?
↓
Keyword?
```

এই Habit

তোমাকে

Average থেকে আলাদা করবে।

Common Beginner Mistakes ⚠

❌ Desktop-এ সব Code রাখা।

❌ File-এর নাম

```
new.c
```

❌ Sample Test ছাড়া Submit।

❌ Compile না করে Run।

❌ Compiler Error না পড়া।

Professional Checklist ✅

Contest-এর আগে

✔ VS Code Ready

✔ Terminal Ready

✔ Folder Ready

✔ Compiler Working

✔ Template Ready

✔ Water Ready 😄

Mini Assignment

Task 1

এই Structure বানাও।

```
CP-Bootcamp/
├── HackerRank/
├── CSES/
├── Codeforces/
├── Contest/
├── Notes/
└── Templates/
```

Task 2

এর ভিতরে

```
Templates/
```

Folder-এ

একটা

```
template.c
```

বানাও।

Task 3

Terminal থেকে

নিজে

Compile করো।

```
gcc template.c -o template
```

Run

```
./template
```

Task 4

`input.txt`

বানাও।

Run করো।

```
./template < input.txt
```

Summary

আজ আমরা শিখলাম—

- Workspace-ও Skill।
- Professional Folder Structure।
- File Naming Convention।
- কেন Compiler ব্যবহার করব।
- Input File দিয়ে Testing।
- Edge Case Testing।
- Contest-এর আগে কী কী Ready রাখতে হবে।

🧠 Reflection

নিজেকে প্রশ্ন করো—

1. আমার Folder Structure কি Professional?
2. আমি কি Terminal ব্যবহার করতে স্বচ্ছন্দ?
3. আমি কি Compiler-এর Error পড়ি, নাকি শুধু Copy-Paste করি?
4. আমি কি নিজের Test Case বানাই?

📖 Mentor's Closing Thought

> **"A messy workspace creates a messy mind. A clean workspace creates a focused programmer."**

আজ Code শেখার আগে আমরা **Battlefield** তৈরি করলাম।
পরের Chapter থেকে আমরা শিখব **কীভাবে একজন Competitive Programmer-এর মতো C লিখতে হয়**, শুধু C code না। 🔥

Day 1 — Chapter 3

Writing C Like a Competitive Programmer

> **"Good programmers write code that works. Great competitive programmers write code that works, is fast to write, and is hard to break."**

🎯 Chapter Goal

এই Chapter শেষে তুমি শিখবে—

- একজন Competitive Programmer-এর Coding Style
- Clean Code vs Contest Code
- Variable Naming
- `main(void)` কেন লিখি?
- `int` vs `long long`
- Input/Output-এর Best Practices
- Common Coding Mistakes
- Professional Habits

Introduction

আজ থেকে আমরা **Code** লিখব।

কিন্তু...

আমি তোমাকে শুধু C শেখাব না।

আমি শেখাব—

> **"কীভাবে একজন Competitive Programmer C লিখে।"**

Lesson 1

Code লেখা আর Code Design করা এক জিনিস না

ধরো একই Problem দুইজন Solve করল।

Student A

```c
#include<stdio.h>

int main(){

int a,b,c;

scanf("%d%d",&a,&b);

c=a+b;

printf("%d",c);

}
```

Student B

```c
#include <stdio.h>

int main(void)
{
    int first_number;
    int second_number;

    scanf("%d %d", &first_number, &second_number);

    int sum = first_number + second_number;

    printf("%d\n", sum);

    return 0;
}
```

দুটোই Accepted।

কিন্তু...

Professional কোনটা?

অবশ্যই দ্বিতীয়টা।

🤔 কেন?

কারণ Code শুধু Compiler পড়ে না।

মানুষও পড়ে।

আর...

সবচেয়ে বেশি পড়বে কে?

Future You.

Mentor Note 💡

৬ মাস পরে নিজের Code খুলে যদি নিজেই না বুঝো...

তাহলে Code Clean না।

Lesson 2

Contest Code vs Production Code

এটা খুব Important।

অনেকে YouTube দেখে বলে—

> Variable-এর নাম এক অক্ষরের হবে।

সবসময়?

না।

Production

```
customer_total_balance
```

Contest

```
balance
```

Production

```
current_temperature
```

Contest

```
temp
```

Production

```
student_total_marks
```

Contest

```
marks
```

Contest-এ

ছোট নাম ব্যবহার করা হয়।

কিন্তু...

অর্থহীন নাম না।

ভুল

```
int a;
int b;
int c;
```

ভালো

```
int sum;
int count;
int max;
int min;
int index;
```

Golden Rule ⭐

Variable-এর নাম এমন হবে...

যাতে Comment না লিখেও বুঝা যায়।

Lesson 3

Indentation

Compiler-এর কাছে

এগুলো একই।

```
if(a>b){
printf("A");
}
```

```
if (a > b)
{
    printf("A");
}
```

কিন্তু...

Human-এর কাছে?

দ্বিতীয়টা।

আজ থেকে

৪ Space।

সবসময়।

না

```
if(a>b)
{
printf("A");
}
```

বরং

```
if (a > b)
{
    printf("A");
}
```

Lesson 4

Braces

অনেকে লেখে

```
if (x)
    printf("YES");
```

আমি Recommend করি না।

সবসময়

```c
if (x)
{
    printf("YES");
}
```

কেন?

কারণ...

আগামীকাল

আরেকটা Line যোগ করলে?

Bug.

Professional Habit

সবসময় Brace.

Lesson 5

main()

আমি লিখি

```c
int main(void)
{
    return 0;
}
```

অনেকে লেখে

```
int main()
```

Difference?

আজকের জন্য

দুটোই Accept.

কিন্তু...

C Language অনুযায়ী

```
int main(void)
```

আরও পরিষ্কারভাবে বোঝায়—

> "এই function কোনো argument নেয় না।"

আমাদের Bootcamp-এ

সবসময়

```
int main(void)
```

Lesson 6

return 0;

অনেকে লিখে না।

আমি লিখি।

কেন?

কারণ...

এটা Program-কে জানায়—

> "সবকিছু Successfully শেষ হয়েছে।"

আজ থেকে

সব Program-এ

```
return 0;
```

Lesson 7

Input Style

ভুল

```
scanf("%d%d",&a,&b);
```

ভালো

```
scanf("%d %d", &a, &b);
```

কেন?

Readable.

Output

ভুল

```
printf("%d",sum);
```

ভালো

```
printf("%d\n", sum);
```

`\n`

কেন?

Contest-এ

কিছু Judge

শেষে Newline Expect করে।

আর Terminal Output-ও Clean থাকে।

Lesson 8

Magic Number

ধরো

```
if(age ⩾ 18)
```

১৮ কেন?

কেউ জানে না।

যদি অনেকবার লাগে

```
#define ADULT_AGE 18
```

Contest-এ

সবসময় দরকার না।

কিন্তু Habit ভালো।

Lesson 9

int vs long long

এই Chapter-এর সবচেয়ে Important অংশ।

অনেকে সব জায়গায়

```
int
```

ব্যবহার করে।

কিন্তু...

Contest-এ

Number হতে পারে

```
1000000000
```

আবার

```
1000000000000
```

`int`

প্রায়

```
-2,147,483,648
↓
2,147,483,647
```

এর বেশি?

Overflow.

`long long`

অনেক বড়।

Rule

যদি Problem-এ

```
10^9
```

পর্যন্ত থাকে

`int` সাধারণত যথেষ্ট।

যদি

```
10^12
10^15
10^18
```

থাকে

```
long long
```

Mentor Tip 💡

Problem Statement-এ

Constraint

দেখে

Data Type নির্বাচন করো।

এটা Guess না।

Decision.

Lesson 10

Global Variable?

না।

এখন না।

সবকিছু

```
main()
```

এর ভিতরে রাখো।

Global Variable

পরে শিখব।

Lesson 11

Comment

ভুল

```
// input

scanf(...)
```

যেটা Code থেকেই বোঝা যায়

সেখানে Comment লাগবে না।

ভালো Comment

```
// Find the maximum element from the array
```

Comment

Explain WHY.

না যে

WHAT.

Lesson 12

One Job, One Variable

ভুল

```
int a;
```

প্রথমে Age.

পরে Sum.

তারপর Count.

Professional

```
int age;
int sum;
int count;
```

এক Variable

এক Meaning.

Lesson 13

White Space

ভুল

```
if(a>b&&b<c)
```

ভালো

```
if (a > b && b < c)
```

Readable.

Lesson 14

The "Compile Before Thinking It's Correct" Rule

অনেক Beginner

Code লিখে...

দেখেও না Compile হয় কিনা।

আজ থেকে Rule

```
Write
↓
Compile
↓
Fix Errors
↓
Run
↓
Test
↓
Submit
```

কখনো

```
Write
↓
Submit
```

না।

Common Beginner Mistakes ⚠

❌

Variable-এর নাম

```
a,b,c,d,e
```

❌

Indentation নেই।

❌

Newline নেই।

❌

সব Data Type

```
int
```

❌

Compiler Warning Ignore.

❌

Brace বাদ।

Professional Coding Checklist ✅

Code শেষ হলে

নিজেকে জিজ্ঞেস করো—

✔ Variable Name বোঝা যাচ্ছে?

✔ Indentation ঠিক?

✔ Brace আছে?

✔ `return 0;` আছে?

✔ `\n` দিয়েছি?

✔ Data Type ঠিক?

✔ Compile Warning আছে?

🏆 Contest Tip

Contest-এ

Beautiful Code

দেখে কোনো Extra Point দিবে না।

কিন্তু...

Beautiful Code

Bug কমায়।

Bug কমলে

Accepted বেশি।

Mini Exercise

নিচের Code-টা Professional Style-এ Rewrite করো।

```
#include<stdio.h>
int main(){
int a,b,c;
scanf("%d%d",&a,&b);
c=a+b;
printf("%d",c);
}
```

Code-এর Logic পরিবর্তন করবে না।

শুধু Style Improve করবে।

Reflection

আজকের শেষে নিজের কাছে প্রশ্ন করো—

১.

আমি কি নিজের Code ৬ মাস পরে পড়তে পারব?

২.

আমার Variable-এর নাম Meaningful?

৩.

আমি কি Compiler-এর জন্য লিখছি...

নাকি Human-এর জন্যও?

📌 Chapter Summary

আজ আমরা শিখলাম—

- একজন Competitive Programmer-এর Coding Style।
- কেন Readability গুরুত্বপূর্ণ।
- `main(void)` ব্যবহার করার কারণ।
- Brace, Indentation, Variable Naming।
- `int` বনাম `long long` ।
- Compile → Run → Test → Submit Workflow।

🧠 Mentor's Closing Thought

আজকের Chapter-এর সবচেয়ে গুরুত্বপূর্ণ শিক্ষা হলো—

> **"Fast code লেখা গুরুত্বপূর্ণ, কিন্তু এমন code লেখা আরও গুরুত্বপূর্ণ যেটা তুমি ১ সপ্তাহ পরে খুলে ১০ সেকেন্ডে বুঝতে পারবে।"**

Day 1 — Chapter 4

Thinking Before Coding

How to Read Any Programming Problem Like a Competitive Programmer

> **"The best programmers don't write code faster. They understand the problem faster."**

🎯 Chapter Goal

এই Chapter শেষ হলে তুমি শিখবে—

- Problem Statement কীভাবে পড়তে হয়।
- কোন লাইন গুরুত্বপূর্ণ আর কোনটা গল্প।
- কীভাবে Problem-কে Programming Language-এ Translate করতে হয়।
- কীভাবে Code লেখার আগেই Algorithm মাথায় তৈরি হয়।
- আমি (তোমার Mentor) কীভাবে Contest-এ Problem পড়ি।

একটি কথা...

এখন পর্যন্ত আমরা শিখেছি—

- CP কী
- Environment
- Coding Style

এখন থেকে **Bootcamp-এর আসল অংশ শুরু।**

এখন আমরা আর C Language শিখব না।

আমরা শিখব **Thinking**।

Lesson 1

The Biggest Beginner Mistake

ধরো Contest-এ ১০টা Problem এসেছে।

তুমি Problem A খুললে।

Beginner কী করে?

```
Problem খুললো

↓

২ লাইন পড়লো

↓

"ওহ, বুঝে গেছি"

↓

VS Code Open

↓

Code
```

Professional কী করে?

```
Problem
↓
Read
↓
Read Again
↓
Underline Keywords
↓
Think
↓
Small Example
↓
Algorithm
↓
Code
```

🧠 Mentor Note

Contest-এ ৩ মিনিট বেশি Problem পড়া, ৩০ মিনিট Debug করার চেয়ে অনেক ভালো।

Lesson 2

Every Problem Has Four Parts

যেকোনো Programming Problem-এ ৪টা অংশ থাকে।

```
Story
↓
Input
↓
Output
↓
Constraints
```

সব Beginner Story পড়ে।

Professional?

Story Skip করে না...

কিন্তু Story থেকে **Problem Extract** করে।

Example

```
Rahim has N boxes.
Each box contains some balls.
Find the box containing the maximum number of balls.
```

Beginner দেখে

```
Rahim
↓
Boxes
↓
Balls
```

Professional দেখে

```
Input
↓
Array
↓
Maximum
```

💡 Mentor Rule

Story হলো Context.

Algorithm Story দেখে না।

Algorithm শুধু Data দেখে।

Lesson 3

The Translation Technique

এটাই Bootcamp-এর সবচেয়ে Powerful Technique.

আজ থেকে...

English → Code না।

English → Logic.

Example

Problem

```
Print the largest number.
```

Translate

```
Largest

↓

Maximum Pattern

↓

Comparison

↓

Traversal

↓

Store Best
```

আরেকটা

```
Count vowels.
```

Translate

```
String
↓
Traversal
↓
Check
↓
Counter
```

আরেকটা

```
Reverse a number.
```

Translate

```
Digit
↓
Last Digit
↓
Build New Number
```

দেখলে?

এখনও Code আসেনি।

Lesson 4

Keywords are Gold

Contest-এ কিছু Keyword দেখলেই মাথায় Pattern আসা উচিত।

যদি লেখা থাকে

```
Maximum

Highest

Largest

Longest
```

Brain

↓

```
Maximum Pattern
```

যদি লেখা থাকে

```
  Minimum

  Smallest

  Lowest

↓

  Minimum Pattern
```

যদি লেখা থাকে

```
  Count

  How many

  Frequency

↓

  Counting
```

যদি লেখা থাকে

```
  Reverse

↓

  Digit / String Reverse
```

যদি লেখা থাকে

```
  Prime

  Divisor

  GCD

  LCM

↓

  Math
```

📝 Mentor Exercise

আজ থেকে Problem পড়ার সময়

Keyword Highlight করবে।

Lesson 5

The Five Questions Rule ⭐⭐⭐⭐⭐

এই Rule পুরো Bootcamp-এর সবচেয়ে Important Rule.

আমি ব্যক্তিগতভাবে এখনও এটা Follow করি।

Problem খুলেই...

নিজেকে এই ৫টা Question করি।

Question 1

Input কী?

```
Single Number?

Array?

String?

Matrix?
```

Question 2

Output কী?

```
Print?

Count?

Maximum?

Minimum?

YES / NO?
```

Question 3

Keyword কী?

```
Prime?

Palindrome?

Reverse?

Maximum?

Sort?
```

Question 4

Pattern কী?

```
Math?

Array?

Simulation?

String?
```

Question 5

সবচেয়ে Simple Solution কী?

এটাই হচ্ছে...

Brute Force.

Mentor Note 💡

Brute Force ভাবতে লজ্জা পেও না।

সব Professional

Brute Force দিয়েই শুরু করে।

Lesson 6

Sample Input Never Lies

অনেকে Sample দেখে না।

এটা বড় ভুল।

Problem

```
Input

5

2 4 6 8 10
```

Output

```
30
```

Professional কী করবে?

মাথায় Dry Run করবে।

```
sum=0
↓
2
↓
6
↓
12
↓
20
↓
30
```

যদি মাথায় না মিলে...

Code লিখবে না।

Lesson 7

Make Your Own Example

এটা Contest Winner-দের Habit.

Sample Input

```
5

1 2 3 4 5
```

Professional আরেকটা বানায়।

```
1

10
```

আরেকটা

```
5

0 0 0 0 0
```

আরেকটা

```
5

-1 -5 -9 -10 -2
```

কেন?

কারণ Sample সব Bug ধরে না।

Lesson 8

Draw Before Code

অনেক Problem

কাগজে আঁকলে সহজ হয়।

ধরো

```
3 8 2 9 1
```

Maximum?

আমি মাথায় লিখি।

```
max=3

↓

8

↓

8

↓

9

↓

9
```

Palindrome?

12321

↓

1

↓

2

↓

3

↓

2

↓

1

Drawing

=

Thinking.

Lesson 9

Algorithm is Just a Story

অনেকে ভাবে

Algorithm মানে বড় কিছু।

না।

Algorithm হচ্ছে...

নিজেকে বাংলা ভাষায় বোঝানো।

Example

```
১.

প্রথম সংখ্যাকে Maximum ধরো।

২.

বাকি সব সংখ্যা Check করো।

৩.

বড় হলে Update করো।

৪.

শেষে Print করো।
```

এই Algorithm.

Code না।

Lesson 10

Never Start with Syntax

Problem দেখে

```
for

while

if
```

ভাববে না।

ভাববে

```
Need Loop

↓

Need Counter

↓

Need Maximum
```

তারপর

Syntax.

Mentor Rule

আগে Logic.

পরে Language.

Common Beginner Mistakes ⚠

❌ পুরো Problem না পড়া।

❌ Story-তে হারিয়ে যাওয়া।

❌ Sample Ignore করা।

❌ Keyword না দেখা।

❌ Brute Force না ভাবা।

❌ Code দিয়ে চিন্তা শুরু করা।

Professional Workflow ⭐

আজ থেকে প্রতিটা Problem-এর জন্য এই Workflow ব্যবহার করবে।

Problem

↓

Read

↓

Highlight Keywords

↓

Find Pattern

↓

Create Example

↓

Brute Force

↓

Algorithm

↓

Code

↓

Dry Run

↓

Submit

🏆 Contest Tip

যদি কোনো Problem বুঝতেই না পারো...

Code লিখবে না।

আবার পড়ো।

অনেক সময়

দ্বিতীয়বার পড়লেই Solution মাথায় আসে।

🧠 Real Example

Problem

```
Find the second largest number.
```

Beginner

```
Second Largest??
```

কঠিন।

Professional

```
Largest জানি।

↓

Second Largest লাগবে।

↓

দুইটা Variable লাগতে পারে।

↓

Comparison.
```

Problem ছোট হয়ে গেল।

Mini Assignment

Code লিখবে না।

শুধু Pattern লিখবে।

Problem 1

```
Find the smallest number in an array.
```

Problem 2

```
Check whether a string is a palindrome.
```

Problem 3

```
Count how many numbers are divisible by 3.
```

Problem 4

```
Print the reverse of an array.
```

Problem 5

```
Find the average of N numbers.
```

Reflection

নিজেকে প্রশ্ন করো—

১.

আমি কি Story পড়ি, নাকি Problem Extract করি?

২.

আমি কি Sample Dry Run করি?

৩.

আমি কি Code-এর আগে Algorithm লিখতে পারি?

📌 Chapter Summary

আজ আমরা শিখলাম—

- Problem পড়ার Professional Method।
- English থেকে Logic-এ Translate করা।
- Keyword Detection।
- Five Questions Rule।
- Brute Force দিয়ে শুরু করা।
- Sample Dry Run।
- নিজের Test Case বানানো।
- Algorithm আগে, Code পরে।

💎 Mentor's Secret

আমি তোমাকে একটা Secret বলি।

তুমি হয়তো ভাবছো...

> "ভালো Competitive Programmer মানে অনেক Algorithm জানে।"

আংশিক সত্য।

কিন্তু আমি যত Senior ICPC Programmer দেখেছি, তাদের একটা Common Habit ছিল—

> **তারা Code দ্রুত লেখে না। তারা Problem দ্রুত বুঝে ফেলে।**

এটাই তাদের Superpower।

Day 1 — Chapter 5

Your First Competitive Programming Problems

The Right First Steps Matter More Than Fast Steps.

> **"Your first 20 problems will shape the way you think for the next 2000."**

🎯 Chapter Goal

এই Chapter শেষে তুমি—

- প্রথমবার Competitive Programming-এর জন্য Problem Solve শুরু করবে।
- বুঝবে **কেন** এই Problem করছি।
- Code না, **Thinking Pattern** শিখবে।
- HackerRank-এ কীভাবে Practice করতে হয় তা শিখবে।

📖 Introduction

আজ পর্যন্ত আমরা শিখেছি—

- CP কী
- Thinking কীভাবে করতে হয়
- Problem কীভাবে পড়তে হয়
- Professional Coding Style

এখন প্রশ্ন...

> **"কোথা থেকে Problem Solve শুরু করব?"**

এখানেই বেশিরভাগ Beginner ভুল করে।

অনেকে প্রথম দিনই Codeforces খুলে।

৩০ মিনিট বসে থাকে।

একটা Problem-ও Solve হয় না।

তারপর ভাবে—

> **"CP আমার জন্য না।"**

না।

Problem তোমার না।

Starting Point ভুল ছিল।

🧠 Mentor Rule #1

Easy Problem Solve করতে লজ্জা পেও না।

অনেক Beginner ভাবে—

> "এটা তো খুব Easy."

Professional ভাবে—

> "আমি কি এটা ২ মিনিটে করতে পারি?"

দুইটা আলাদা চিন্তা।

ধরো...

একটা Problem Solve করতে তোমার

২০ মিনিট লাগছে।

একই Problem

ICPC Team

২ মিনিটে করছে।

Difference?

Knowledge না।

Practice.

🎯 Warm-up Roadmap

আজকে আমরা HackerRank-এর

Introduction Section শেষ করব।

কেন?

কারণ এখানে

CP-এর জন্য দরকারি Basic Habit তৈরি হবে।

Problem 1

Say "Hello, World!" in C

Difficulty

⭐☆☆☆☆

Link

HackerRank → C → Introduction → Hello World

Goal

এটা Programming শেখার জন্য না।

এটা Environment Test.

Skill

- VS Code ঠিক আছে?
- GCC কাজ করছে?
- Submit করতে পারছো?

Pattern

```
Output
```

Expected Time

২ মিনিট

Common Mistake

- Semicolon
- Compile না করা
- Wrong Output

Mentor Note

এই Problem Ignore করবে না।

Contest-এর আগে Environment Check-এর মতো।

Problem 2

Playing With Characters

Goal

Character Input.

Skill

```
char

scanf

printf
```

Pattern

```
Input → Output
```

Expected Time

৫ মিনিট

Common Mistake

```
%c
```

এবং

```
%s
```

Mix করে ফেলা।

Problem 3

Sum and Difference of Two Numbers

Goal

Integer

Float

Formatting

Skill

```
Basic Arithmetic
```

Pattern

```
Math
```

Expected Time

৫ মিনিট

Common Mistake

Format Specifier.

Problem 4

Functions in C

Goal

Function Call.

Pattern

```
Function
```

Skill

Problem Break করা।

Expected Time

১০ মিনিট

Mentor Note

Contest-এ

সব Problem Function দিয়ে করতে হবে এমন না।

কিন্তু

Function-এ চিন্তা করতে শিখবে।

Problem 5

Pointer in C

Goal

Pointer-এর Basic।

Pattern

```
Pointer
```

Expected Time

১৫ মিনিট

Important

Pointer মুখস্থ করবে না।

Concept বুঝবে।

Problem 6

Conditional Statements in C

Goal

if-else

Pattern

Decision Making

Skill

Branching

Common Mistake

Boundary.

Problem 7

For Loop in C

Goal

Iteration.

Pattern

Loop

Skill

Loop Visualization.

Mentor Note

Loop CP-এর Heart.

Problem 8

Sum of Digits of a Five Digit Number

Goal

Digit Manipulation.

Pattern

```
Math

+

Digit
```

Skill

```
%

/

Operator
```

Important

এটা CP-তে অনেকবার আসবে।

Problem 9

Bitwise Operators

Goal

Bitwise-এর Basic Idea.

Expected

সমাধান করতে পারলে ভালো।

না পারলেও Problem নেই।

আমরা পরে আবার আসব।

Problem 10

Printing Pattern Using Loops

Goal

Nested Loop.

Pattern

Simulation.

Skill

Visualization.

🧠 Problem Solving Routine

প্রতিটা Problem-এর জন্য একই Routine.

Step 1

পুরো Statement পড়ো।

Step 2

নিজেকে জিজ্ঞেস করো।

```
Input?

↓

Output?

↓

Pattern?

↓

Keyword?
```

Step 3

Example Dry Run.

Step 4

Algorithm বাংলায় লিখো।

Step 5

Code.

Step 6

নিজের Test Case.

Step 7

Submit.

Step 8

Accepted?

YES

↓

Next.

NO

↓

Read Error.

↓

Fix.

↓

Submit Again.

⚠ Don't Do This

Problem খুলে

YouTube Search.

❌

Problem খুলে

ChatGPT

"Code দাও"

❌

Problem খুলে

Editorial.

❌

আজ থেকে Rule.

```
20 Minutes
↓
নিজে ভাবো
↓
Hint
↓
আরও ভাবো
↓
তারপর Editorial
```

💡 The 20-Minute Rule

যদি

২০ মিনিট

কোনো Idea-ই না আসে...

তাহলে Hint দেখো।

যদি

৩০ মিনিট

পেরিয়ে যায়...

Editorial পড়ো।

কিন্তু...

Copy করবে না।

Editorial বন্ধ করো।

নিজে আবার লিখো।

🏆 Contest Habit

Accepted হওয়ার পরে

Problem Close করবে না।

নিজেকে প্রশ্ন করবে।

```
আমি কী শিখলাম?

Pattern কী?

আরও Short করা যেত?

Bug কোথায় ছিল?
```

📓 Bootcamp Notebook

আজ থেকে

প্রতিটা Problem-এর জন্য

এই Template ব্যবহার করবে।

```
# Problem

## Link

## Pattern

## Difficulty

## My First Thought

## Algorithm

## Mistake

## Lesson Learned
```

🎯 Today's Mission

আজকের Target

Problem	Target Time
1	2 min
2	5 min
3	5 min
4	10 min
5	15 min
6	10 min
7	10 min
8	15 min

Problem	Target Time
9	20 min (optional)
10	20 min

✅ Success Criteria

আজকের শেষে তুমি যদি—

- সব Problem Solve করতে না-ও পারো,
- কিন্তু প্রতিটা Problem-এর Pattern চিনতে পারো,
- নিজের Algorithm বাংলায় লিখতে পারো,
- Code করার আগে ৫ মিনিট ভাবতে পারো,

তাহলে Day 1 সফল।

💎 Mentor's Secret

আজ একটা জিনিস মনে রাখবে।

অনেকে ভাবে...

> **Competitive Programming = অনেক Problem Solve করা।**

আমি বলি...

> **Competitive Programming = একই Problem থেকে যত বেশি শেখা যায়।**

একজন Beginner

১০টা Problem Solve করে।

একজন Professional

একটা Problem Solve করে

১০টা Lesson শিখে।

📌 Day 1 Progress

- ✅ Chapter 1 — Mindset
- ✅ Chapter 2 — Battlefield Setup
- ✅ Chapter 3 — Professional C Style
- ✅ Chapter 4 — Thinking Before Coding
- ✅ Chapter 5 — First Problems

Day 1 প্রায় ৬০% শেষ।

Day 1 — Chapter 6

How to Solve a Problem Without Looking at the Solution

The Mentor Method – Learn How to Think, Not What to Type

> **"A Competitive Programmer doesn't search for code. He searches for the idea."**

🎯 Chapter Goal

এই Chapter Bootcamp-এর সবচেয়ে গুরুত্বপূর্ণ Chapter-গুলোর একটি।

আজ আমি তোমাকে Code শেখাব না।

আমি তোমাকে **আমার Brain-এর ভিতরে নিয়ে যাব।**

মানে...

একটা Problem পেলে **আমি কীভাবে চিন্তা করি**, সেটা দেখাব।

আজকের পর থেকে তোমার Problem দেখার দৃষ্টিভঙ্গি বদলে যাবে।

Introduction

ধরো...

তুমি HackerRank-এ একটা Problem খুললে।

Beginner কী করে?

```
Problem
↓
২ মিনিট ভাবলো
↓
পারলাম না
↓
Google
↓
YouTube
↓
ChatGPT
↓
Copy
↓
Accepted
```

সে Accepted পেল।

কিন্তু...

সে **Programming শেখেনি।**

Professional কী করে?

```
Problem
↓
Read
↓
Think
↓
Draw
↓
Brute Force
↓
Improve
↓
Algorithm
↓
Code
```

💡 Mentor Rule

Programming-এ Accepted পাওয়া Success না।

Accepted-এর আগে তুমি কীভাবে চিন্তা করলে, সেটাই Success।

আজ আমরা একটা Problem Solve করব

না...

Code লিখে না।

Brain দিয়ে।

Problem

```
Given N numbers,

Print the largest number.
```

Step 1

Don't Think About Code

সবচেয়ে প্রথম Rule।

Code ভাবা নিষেধ।

তুমি যদি এখন ভাবো—

```
for(...)
```

Stop.

আমি প্রথমে শুধু Problem-টা দেখি।

Step 2

Rewrite the Problem

Professional Programmer Problem-কে নিজের ভাষায় লিখে।

Original

```
Given N numbers,

Print the largest number.
```

আমার Brain-এ এটা হয়—

```
আমাকে অনেকগুলো Number দেওয়া হবে।

সবগুলোর মধ্যে সবচেয়ে বড়টা বের করতে হবে।
```

এখন Problem অনেক সহজ।

Mentor Note 💡

যদি Problem নিজের ভাষায় Explain করতে না পারো...

তাহলে Problem এখনও বুঝোনি।

Step 3

Make a Small Example

আমি কখনো Sample Input দিয়ে শুরু করি না।

আমি নিজের Example বানাই।

```
3 8 2 9 1
```

Question

সবচেয়ে বড়?

```
9
```

Easy.

এখন আমি আরেকটা বানাই।

```
-5 -2 -9
```

Answer?

```
-2
```

আরেকটা

7

Answer?

7

এখন আমার Brain বুঝতে শুরু করেছে।

কেন আমি নিজের Example বানালাম?

কারণ...

Problem Statement তোমাকে Answer বলে না।

Example বলে।

Step 4

Observe

এখন আমি Observe করি।

3

↓

8

↓

2

↓

9

↓

1

আমি কী করলাম?

3

↓

8 বড়

↓

8

↓

9 বড়

↓

9

হুম...

আমি একটা Pattern দেখলাম।

আমি সব Number Compare করছি।

Step 5

Name the Pattern

Professional Programmer Pattern-এর নাম দেয়।

এটা কী?

Sorting?

না।

Searching?

না।

Math?

না।

Pattern

```
Maximum
```

যেদিন Pattern চিনতে শিখবে...

সেদিন থেকেই CP সহজ লাগবে।

Step 6

Think Brute Force

এখন আমি সবচেয়ে Simple Solution ভাবি।

সবগুলো Number Check করব।

Question

Correct?

হ্যাঁ।

Fast?

জানি না।

আগে Correct.

পরে Fast.

Mentor Rule

> **Correctness > Cleverness**

অনেক Beginner Smart হতে গিয়ে Wrong Answer পায়।

Professional আগে Correct Solution বানায়।

Step 7

Write the Algorithm in Bangla

এখনও Code না।

বাংলায় লিখি।

```
১।

প্রথম Number-কে Largest ধরো।

↓

২।

বাকি সব Number Check করো।

↓

৩।

যদি Current Number বড় হয়

↓

Largest Update করো।

↓

৪।

শেষে Largest Print করো।
```

Algorithm Ready.

Step 8

Dry Run Before Coding

Algorithm ঠিক?

চলো Test করি।

```
Input

3 8 2 9 1
```

Step

```
Largest = 3
```

Next

```
8

↓

Update
```

Largest

```
8
```

Next

```
2
↓
No Update
```

Next

```
9
↓
Update
```

Next

```
1
↓
No
```

Answer

```
9
```

Algorithm ঠিক।

Step 9

NOW Write Code

এখন Code.

এখন।

তার আগে না।

🧠 Mentor Secret

তুমি কি লক্ষ্য করেছো?

আমরা এতক্ষণে...

- Problem বুঝেছি
- Example বানিয়েছি
- Pattern চিনেছি
- Algorithm লিখেছি
- Dry Run করেছি

কিন্তু এক লাইন Code লিখিনি।

Professional Programming এমনই।

এবার আরেকটা Example

Problem

```
Count Even Numbers
```

Beginner

```
Loop?
```

Professional

```
Need
↓
Counter
↓
Loop
↓
Check Even
```

দেখলে?

সে Syntax ভাবছে না।

সে Logic ভাবছে।

আরেকটা Example

Problem

```
Check Prime
```

Professional

```
Need
↓
Math
↓
Divisor
↓
Check
```

Code পরে।

Lesson

Never Marry Your First Idea

এটা Senior Programmer-এর Habit।

ধরো

তোমার মাথায় Solution আসল।

Stop.

নিজেকে প্রশ্ন করো।

```
এর চেয়ে সহজ উপায় আছে?
```

সবসময় না।

কিন্তু Question করবে।

Three-Level Thinking

এখন থেকে

প্রতিটা Problem-এ

এই তিনটা Level ভাববে।

Level 1

Problem কী চায়?

Level 2

সবচেয়ে Simple Solution কী?

Level 3

আরও ভালো Solution আছে?

এটাই Algorithm Design.

⚠ Common Beginner Mistakes

❌

২ মিনিটে Solution না পেয়ে হতাশ।

❌

Editorial খুলে ফেলা।

❌

নিজের Example না বানানো।

❌

Algorithm না লিখে Code।

❌

Dry Run না করা।

Mentor's Problem Solving Checklist ⭐

আজ থেকে প্রতিটা নতুন Problem-এর জন্য এই Checklist ব্যবহার করবে।

- ☐ পুরো Problem পড়েছি?
- ☐ নিজের ভাষায় Explain করতে পারি?
- ☐ নিজের Example বানিয়েছি?
- ☐ Pattern চিনেছি?
- ☐ Brute Force ভেবেছি?
- ☐ বাংলায় Algorithm লিখেছি?
- ☐ Dry Run করেছি?
- ☐ এখন Code লিখব।

🏆 Contest Tip

Contest-এ যদি কোনো Problem ১০ মিনিটেও বুঝতে না পারো...

Code লিখো না।

আবার পড়ো।

আরও একটা Example বানাও।

অনেক সময় Solution

Problem Statement-এর মধ্যেই লুকিয়ে থাকে।

🎯 Today's Exercise

Code লিখবে না।

শুধু নিচের Problem-গুলোর জন্য

এই ৪টা জিনিস লিখবে।

Problem 1

```
Find the minimum number in an array.
```

লিখবে—

- নিজের ভাষায় Problem
- Pattern
- Brute Force
- Algorithm

Problem 2

```
Count odd numbers.
```

একইভাবে।

Problem 3

```
Reverse a string.
```

একইভাবে।

💎 Mentor's Golden Rule

আমি তোমাকে একটা Rule দিচ্ছি।

এই Rule আমি চাই তুমি GitHub-এর README-তেও লেখো।

> **Never ask "How do I code this?"**
>
> **Ask "How do I think about this?"**

এই এক Line...

তোমার Programming Career বদলে দিতে পারে।

📌 Chapter Summary

আজ আমরা শিখলাম—

- Code লেখার আগে কীভাবে ভাবতে হয়।
- Problem-কে নিজের ভাষায় Rewrite করা।
- নিজের Example বানানোর গুরুত্ব।
- Pattern চেনা।
- Brute Force দিয়ে শুরু করা।
- বাংলায় Algorithm লেখা।
- Dry Run-এর পরে Code লেখা।
- তিন ধাপে Problem নিয়ে চিন্তা করা।

🔥 Mentor's Challenge

আজ থেকে Bootcamp-এর বাকি প্রতিটি Problem-এ তুমি আমাকে **Code চাইবে না।**

তুমি আগে আমাকে পাঠাবে—

1. **Problem-এর Pattern**
2. **তোমার Brute Force Idea**
3. **বাংলায় Algorithm**

তারপর আমি Review করব।

কারণ...

আমি চাই না তুমি আমার Code লিখো।

আমি চাই, একদিন আমি তোমার Code Review করি।

📖 শেষ কথা

আজকের Chapter-এর সবচেয়ে গুরুত্বপূর্ণ বাক্যটি আবার লিখছি—

> **"একজন Beginner Code লেখে। একজন Competitive Programmer প্রথমে চিন্তা করে, তারপর Algorithm বানায়, তারপর Code লেখে।"**

যেদিন এই Habit তোমার Default হয়ে যাবে, সেদিন থেকে তুমি আর শুধু C Programmer থাকবে না—**তুমি একজন Problem Solver হয়ে উঠবে।** 🚀

Day 1 — Chapter 7

The First Real Practice Session

Mentor Mode – We Don't Write Code. We Build Thinking.

> **"Today you stop being a C learner. Today you become a Problem Solver."**

🎯 Chapter Goal

আজ প্রথমবার আমরা **বাস্তব Competitive Programming Practice** করব।

কিন্তু...

আমরা অন্যদের মতো করব না।

আমরা Code দিয়ে শুরু করব না।

আমরা **Brain দিয়ে শুরু করব।**

📌 Mentor's Rule

আজ থেকে আমাদের Bootcamp-এর সবচেয়ে গুরুত্বপূর্ণ Rule শুরু।

Rule #1

```
Problem খুলে

↓

Code না।

↓

Think.
```

Rule #2

তুমি যদি আমাকে কখনো বলো

> "Code দাও"

আমি Code দিব না।

আমি আগে জিজ্ঞেস করব—

> "তুমি কী ভাবছো?"

কারণ...

আমি তোমাকে মাছ দিতে চাই না।

আমি মাছ ধরা শেখাতে চাই।

🧠 The Mentor Method

আজ থেকে প্রতিটা Problem এই ১০টা Step-এ Solve হবে।

```
Problem
↓
Read
↓
Highlight
↓
Pattern
↓
Own Example
↓
Brute Force
↓
Algorithm
↓
Dry Run
↓
Code
↓
Review
```

এই Flow পুরো Bootcamp-এ একটুও পরিবর্তন হবে না।

🔥 আজকের প্রথম Problem

আমরা শুরু করব HackerRank-এর সবচেয়ে সহজ Problem দিয়ে।

Problem

Playing With Characters

(তুমি HackerRank-এ খুলবে। এখন Code দেখবে না।)

Step 1

Read

Problem পুরোটা একবার পড়ো।

তারপর আবার পড়ো।

এখনও Code না।

Step 2

Highlight

Problem পড়ে গুরুত্বপূর্ণ Keyword বের করো।

ধরো Problem-এ আছে—

```
Character

Word

Sentence
```

Highlight.

কারণ এগুলোই Input.

Step 3

Input Analysis

নিজেকে প্রশ্ন করো।

```
কতগুলো Input?

Type কী?

char?

string?
```

এখনও Code না।

Step 4

Output Analysis

নিজেকে প্রশ্ন করো।

```
Print করতে হবে?

নাকি Return?

এক লাইনে?

আলাদা লাইনে?
```

Step 5

Pattern

এখন Pattern চিনো।

```
Input

↓

Output
```

এটা কি

Math?

না।

Array?

না।

Pattern হলো—

```
Basic Input / Output
```

Mentor Note 💡

সব Problem-এ বড় Algorithm লাগে না।

অনেক Problem-এর লক্ষ্য

Language Check.

Step 6

Make Your Own Example

Problem-এর Sample ছাড়াও নিজের Example বানাও।

```
A

Hello

Programming
```

নিজেকে জিজ্ঞেস করো—

Expected Output কী?

আরেকটা বানাও।

```
Z

Bangladesh

CP
```

Professional Programmer

সবসময়

নিজের Example বানায়।

Step 7

Brute Force

এখানে Brute Force কী?

খুব Simple।

```
Input নাও

↓

Print করো
```

Correct?

হ্যাঁ।

Fast?

হ্যাঁ।

তাহলে এটাই Final.

সব Problem Optimize করতে হয় না।

Step 8

Algorithm

বাংলায় লিখো।

```
১।

একটা Character Input নাও।

↓

২।

একটা Word Input নাও।

↓

৩।

একটা Sentence Input নাও।

↓

৪।

ঠিক যেভাবে Input এসেছে

সেভাবেই Print করো।
```

Step 9

Dry Run

নিজের মাথায় চালাও।

```
Input

A

Hello

Programming
```

মাথায় Follow করো।

```
Character

↓

Word

↓

Sentence

↓

Print
```

সব ঠিক?

তাহলে Code.

Step 10

Code

এখন।

এখনই।

এর আগে না।

🎯 Mentor Challenge

এখন একটা Challenge।

তুমি Code লেখার আগে

আমাকে এই ৪টা জিনিস লিখে পাঠাবে।

```
Pattern:

Own Example:

Algorithm:

Possible Mistake:
```

Example

```
Pattern

Input / Output

---

Own Example

B

World

Bangladesh

---

Algorithm

Input

↓

Print

---

Possible Mistake

Sentence Input ঠিকভাবে না নেওয়া।
```

আমি Review করব।

তারপর Code।

🧠 Why This Method Works

ধরো

একজন Student

১০০টা Code মুখস্থ করল।

আরেকজন

২০টা Problem-এর Thinking শিখল।

এক বছর পরে

কে বেশি Problem Solve করবে?

দ্বিতীয়জন।

কারণ...

Programming Language বদলায়।

Thinking বদলায় না।

⚠ Common Beginner Mistakes

❌

Statement পুরো না পড়া।

❌

নিজের Example না বানানো।

❌

Code দিয়ে Thinking শুরু করা।

❌

Pattern না দেখা।

❌

Accepted হলেই Problem Close করা।

🏆 Contest Habit

Accepted হওয়ার পরে

নিজেকে জিজ্ঞেস করবে—

```
এই Problem
আবেকবার দিলে
আমি
৫ মিনিটে
করতে পারব?
```

যদি উত্তর

```
না
```

হয়...

Problem এখনও শেখা হয়নি।

💡 Mentor Secret

তুমি জানো?

ICPC World Finalist-রা

Accepted পাওয়ার পরও

Editorial পড়ে।

কেন?

কারণ তারা জানতে চায়—

> **"আমার চেয়ে আরও ভালো Idea ছিল?"**

এই Habit-টাই Great Programmer বানায়।

📓 Bootcamp Notebook

আজ থেকে নতুন Section যোগ করো।

```
# Problem Reflection

## Problem Name

## Time Taken

## Pattern

## First Idea

## Final Idea

## Mistakes

## One Thing I Learned
```

🎯 Today's Assignment

আজ তুমি **HackerRank-এর প্রথম ৩টা Problem** করবে।

1. Hello World
2. Playing With Characters
3. Sum and Difference of Two Numbers

কিন্তু...

নতুন Rule অনুযায়ী।

প্রতিটা Problem-এর জন্য

Code লেখার আগে

Notebook-এ লিখবে—

```
Pattern

↓

Own Example

↓

Algorithm

↓

Possible Mistake
```

তারপর Code।

🧠 Reflection

নিজেকে প্রশ্ন করো—

১।

আজ কি আমি Code-এর আগে ভেবেছি?

২।

আমি কি নিজের Example বানিয়েছি?

৩।

আমি কি Problem-এর Pattern ধরতে পেরেছি?

৪।

আমি কি Algorithm বাংলায় লিখেছি?

📌 Chapter Summary

আজ থেকে Bootcamp-এর আসল Training শুরু।

আমরা শিখলাম—

- Mentor Method
- ১০ ধাপের Problem Solving Framework
- Pattern → Example → Algorithm → Code
- Problem Reflection
- Competitive Programming Notebook ব্যবহার

⚔️ Mentor's Challenge

আজ আমি তোমাকে একটা Promise করতে বলছি।

নিচের Rule-টা GitHub Repository-র README-তে লিখে রাখবে।

> **"I will never copy a solution before I fail honestly."**

কারণ...

একটা Honest Failure,

দশটা Copy-Paste Accepted-এর চেয়ে অনেক বেশি শেখায়।

Day 1 — Chapter 8 (Final Chapter)

Day 1 Mission Complete

Today You Didn't Learn C. You Learned How to Think.

> **"The first day of training doesn't make you strong. It changes the way you train."**

🎯 Chapter Goal

এই Chapter শেষে তুমি—

- Day 1-এ কী শিখলে সেটা Review করবে।
- নিজের Progress Evaluate করবে।
- Day 2-এর জন্য Ready হবে।
- একজন Competitive Programmer-এর Daily Habit তৈরি করবে।

🌅 Congratulations

আজকে...

তুমি C-এর নতুন Syntax শেখোনি।

তুমি নতুন Algorithm-ও শেখোনি।

কিন্তু...

তুমি একটা **Programming Mindset** তৈরি করেছো।

এটাই Bootcamp-এর সবচেয়ে গুরুত্বপূর্ণ Step।

🧠 আজকের Journey

সকালে যদি তোমাকে আমি বলতাম—

> **"Competitive Programming কী?"**

তুমি হয়তো বলতে—

> "Programming Contest."

এখন?

তুমি জানো...

Competitive Programming মানে—

```
Problem
↓
Pattern
↓
Algorithm
↓
Code
↓
Accepted
```

এবং...

Code হলো শেষ Step।

📚 Day 1 Knowledge Map

আজ আমরা যা শিখলাম—

```
Mindset
↓
Workspace
↓
Coding Style
↓
Problem Reading
↓
Thinking
↓
First Practice
↓
Reflection
```

এটাই Day 1.

🧩 Mentor's Biggest Lesson

আজকের পুরো Chapter-গুলোর মধ্যে

সবচেয়ে Important Concept কোনটা?

অনেকে বলবে—

> Variable Naming

না।

অনেকে বলবে—

> Folder Structure

না।

অনেকে বলবে—

> HackerRank

না।

আমি বলব...

> **Thinking Before Coding**

এটাই পুরো Bootcamp-এর Foundation.

🏆 The 10 Golden Rules

আজ থেকে Contest শেষ না হওয়া পর্যন্ত

এই Rules Follow করবে।

Rule 1

```
Read Twice.
```

একবার না।

দুইবার।

Rule 2

```
Code পরে।

Think আগে।
```

Rule 3

নিজের Example বানাও।

Rule 4

Pattern চিনো।

Rule 5

Brute Force দিয়ে শুরু করো।

Rule 6

বাংলায় Algorithm লেখো।

Rule 7

Dry Run করো।

Rule 8

তারপর Code.

Rule 9

Accepted-এর পর Reflection.

Rule 10

কখনো Solution Copy করবে না।

⭐ The Bootcamp Workflow

আজ থেকে

প্রতিটা Problem-এর জন্য

একই Workflow.

Read

↓

Highlight

↓

Pattern

↓

Own Example

↓

Brute Force

↓

Algorithm

↓

Dry Run

↓

Code

↓

Test

↓

Submit

↓

Reflection

এই Workflow-টাই

তোমার Habit হবে।

📝 Self Evaluation

নিজেকে Honest Score দাও।

Problem Reading

⭐⭐⭐⭐⭐

Pattern চিনা

⭐⭐⭐⭐⭐

Algorithm লেখা

⭐⭐⭐⭐⭐

Dry Run

⭐⭐⭐⭐⭐

Coding Discipline

⭐⭐⭐⭐⭐

যেখানে

৩-এর নিচে Score

সেখানে Improvement লাগবে।

🎯 Day 1 Checklist

আজকের শেষে

এই Checklist Follow করো।

Mindset

- [ ] Code আগে লিখি না।
- [ ] Pattern খুঁজি।
- [ ] নিজের Example বানাই।

Coding

- [ ] Clean Code লিখি।
- [ ] Variable-এর Meaning আছে।
- [ ] Brace ব্যবহার করি।
- [ ] `return 0;` লিখি।

Thinking

- [ ] Algorithm বাংলায় লিখতে পারি।
- [ ] Dry Run করি।
- [ ] Sample Analyze করি।

Practice

- [ ] HackerRank Account Ready।
- [ ] Codeforces Account Ready।
- [ ] Folder Structure Ready।

🧠 Mistake Tracker

আজ থেকেই একটা File বানাও।

```
Mistakes.md
```

প্রতিদিন

শুধু Mistake লিখবে।

Example

```
# Mistake Log

---

## Day 1

Problem:

Playing With Characters

Mistake:

scanf() দিয়ে sentence ঠিকভাবে নিতে পারিনি।

Lesson:

Character Input আর String Input আলাদা।
```

Mentor Note

ভালো Programmer

তার Success Track করে না।

তার Mistake Track করে।

📓 Reflection Journal

প্রতিদিন রাতে

এই ৫টা Question-এর Answer লিখবে।

১

আজ আমি কী নতুন শিখলাম?

২

আজ আমার সবচেয়ে বড় Mistake কী?

৩

আজ কোন Problem আমাকে সবচেয়ে বেশি ভাবিয়েছে?

৪

আজ কোন Pattern শিখলাম?

৫

আগামীকাল কী Improve করব?

📊 Bootcamp Progress

```
Day 0

🟥🟥🟥🟥🟥🟥🟥🟥

100%

✅

Day 1

🟥🟥🟥🟥🟥🟥🟥🟥

100%

✅

Day 2

⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜

0%
```

🏹 Contest Mindset

আজ থেকে

Contest-এ গেলে

নিজেকে বলবে না—

> **"আমি জিততে এসেছি।"**

বলবে—

> **"আমি যতগুলো Problem পারি Solve করতে এসেছি।"**

Mindset বদলালে

Pressure কমে।

Performance বাড়ে।

💡 Mentor's Advice

তুমি Security থেকে এসেছো।

এটা তোমার Advantage.

কারণ...

তুমি

Already

- Logic Trace করো।
- Edge Case ভাবো।
- Dry Run করো।
- Bug খুঁজো।

এখন

শুধু Pattern চিনতে শিখতে হবে।

📚 Homework

আজকের Homework

Code না।

Reflection.

Task 1

Bootcamp Notes Update করো।

Task 2

Mistakes.md বানাও।

Task 3

Problem Template বানাও।

Task 4

আজকের ১০টা Golden Rule

হাতে লিখো।

হ্যাঁ।

হাতে।

কেন?

কারণ

হাতে লেখা জিনিস

Brain-এ অনেক বেশি দিন থাকে।

🚀 What's Next?

আগামীকাল

Day 2.

কিন্তু...

এখানে একটা Surprise আছে।

অনেকে ভাববে

Day 2 = Arrays.

না।

আমাদের Bootcamp-এ

Day 2-এর নাম হবে—

"Learning to See Arrays, Not Just Use Arrays."

কারণ...

Beginner Array দেখে—

```
int arr[100];
```

Professional দেখে—

```
Collection
↓
Traversal
↓
Maximum
↓
Minimum
↓
Frequency
↓
Pattern
```

এটাই Difference.

🏅 Day 1 Graduation

আজ থেকে

তুমি শুধু

C শেখা Student না।

আজ থেকে

তুমি

Problem Solving Student.

💎 Mentor's Final Words

আমি তোমাকে Day 1-এর শেষে একটা Quote দিয়ে যেতে চাই।

> **Programming is not about typing fast.**

> **Programming is not about memorizing syntax.**

> **Programming is not about solving 1000 problems.**

Programming is about...

Thinking clearly.

আর Competitive Programming হলো...

Thinking clearly, under pressure.

📖 End of Day 1

```
📘 CP Bootcamp
──────────────────────────

Day 0  ✅ Completed

Day 1  ✅ Completed

Day 2  🔒 Locked

──────────────────────────
```

🎁 Mentor's Gift (শুধু তোমার জন্য)

রাফি, Day 1 শেষ করে আমি একটা জিনিস বলতে চাই।

আমি ইচ্ছা করেই Day 1-এ **Array শেখাইনি, Loop শেখাইনি, Codeforce শুরু করাইনি।**

কারণ আমি একটা Pattern দেখেছি।

গত কয়েক বছর ধরে আমি যত Beginner-কে CP শুরু করতে দেখেছি, তাদের বেশিরভাগই **Code শিখেছে, কিন্তু Problem Solve করা শেখেনি।**

আমি চাই আমাদের Bootcamp আলাদা হোক।

আমি চাই ১৬ দিন পরে তুমি শুধু বলতে না পারো—

> "আমি ১০০টা Problem Solve করেছি।"

আমি চাই তুমি বলতে পারো—

> **"আমি এখন একটা নতুন Problem দেখে কীভাবে চিন্তা শুরু করতে হয়, সেটা জানি।"**

এটাই এই Bootcamp-এর আসল Success।

📌 Day 1 সম্পর্কে আমার নিজের Review

এখন Mentor হিসেবে নিজেরও একটা Review দিচ্ছি।

আমি Day 1-কে যদি ১০০-তে Score দিই, আমি নিজে **৭৮/১০০** দেব।

কারণ এটা Foundation হিসেবে ভালো হয়েছে, কিন্তু একটা জিনিস আমি ইচ্ছা করে Day 1-এ রাখিনি—

বাস্তব Problem Walkthrough।

আর ঠিক এই কারণেই...

Day 2 আমাদের Bootcamp-এর সবচেয়ে গুরুত্বপূর্ণ Day হতে যাচ্ছে।

সেখানে প্রথমবার আমরা **একটা Problem শুরু থেকে Accepted পর্যন্ত** একসাথে করব।

Code না মুখস্থ করে...

চিন্তা করতে শিখে। 🔥