🚀 CP Bootcamp 2026 — Day 6

Theme: Functions + Modular Thinking

Main Goal

এতদিন তুমি:

```
Problem
 ↓
Pattern
 ↓
Code
```

এই flow-তে ছোট problem solve করেছো।

এখন শিখবে:

```
Big Problem
      ↓
Break into Small Parts
      ↓
Write Reusable Functions
      ↓
Combine Solutions
```

Competitive Programming-এ বড় code manage করার foundation এখান থেকেই শুরু।

📖 Day 6 Chapter Map

```
Day 6
 │
 ├── Chapter 1 — Why Functions Matter in CP
 │
 ├── Chapter 2 — Function Mental Model
 │
 ├── Chapter 3 — Function Syntax in C
 │
 ├── Chapter 4 — Parameters & Arguments
 │
 ├── Chapter 5 — Return Value Pattern
 │
 ├── Chapter 6 — Void Function Pattern
 │
 ├── Chapter 7 — Breaking Problems into Functions
 │
 ├── Chapter 8 — Common Function Mistakes
 │
 └── Chapter 9 — Day 6 Assignment & Practice
```

Chapter 1 — Why Functions Matter in CP

Problem Without Function

ধরো:

তোমাকে ৩ বার করতে হবে:

- Maximum বের করা
- Minimum বের করা

- Sum বের করা

Beginner code:

```
// Maximum code

// Minimum code

// Sum code
```

Problem:

- Code repeat হচ্ছে
- Bug হওয়ার chance বাড়ছে
- বড় problem handle করা কঠিন

Function Idea

Function মানে:

> একটি নির্দিষ্ট কাজ করার জন্য আলাদা code block।

Example:

```
Main Problem
      |
      |
  ┌───┼───┐
  │   │   │
max() sum() check()
```

প্রতিটি function:

```
একটা কাজ করবে
```

CP Mental Model

Professional programmer ভাবে:

```
Problem
   |
   |
Break Into Tasks
   |
   |
Create Functions
   |
   |
Combine
```

Example:

Problem:

> একটি array-এর largest prime number বের করো।

Break:

```
1. Prime check করতে হবে

2. Array traverse করতে হবে

3. Maximum track করতে হবে
```

Functions:

```
isPrime()

findMaximum()
```

Chapter 2 — Function Mental Model

একটা function-এর ৩টা অংশ:

```
Function
│
├── Input
│
├── Work
│
└── Output
```

Example:

Maximum function:

Input:

```
10 5
```

Work:

```
Compare
```

Output:

```
10
```

Diagram:

```
Input
  ↓
┌──────────┐
│ Function │
└──────────┘
  ↓
Output
```

Function কেন দরকার?

1. Reuse

একবার লিখবে:

```
max()
```

অনেক জায়গায় use করবে।

2. Readability

Before:

```
100 lines main()
```

After:

```
main()

solve()

checkPrime()

sortArray()
```

3. Debugging

Problem হলে:

```
Prime function
```

চেক করো।

পুরো program না।

Chapter 3 — Function Syntax in C

Basic structure:

```
return_type function_name(parameters)
{
    // code

    return value;
}
```

Example:

```
int add(int a, int b)
{
    return a + b;
}
```

Break:

```
int
↓
Return Type


add
↓
Function Name


a,b
↓
Parameters
```

Calling Function

Function:

```
int add(int a, int b)
{
    return a+b;
}
```

Call:

```
int result;

result = add(5,3);
```

Output:

```
8
```

Complete Program

```
#include <stdio.h>

int add(int a, int b)
{
    return a+b;
}


int main()
{
    int result;

    result = add(5,3);

    printf("%d", result);

    return 0;
}
```

Execution Flow

Computer ভাবে:

```
main()
  |
  |
  call add(5,3)
  |
  |
add()
  |
  |
return 8
  |
  |
main continues
```

Chapter 4 — Parameters & Arguments

এই দুইটা beginner-রা confuse করে।

Parameter

Function তৈরি করার সময়:

```
int add(int a, int b)
```

এখানে:

```
a,b = Parameters
```

Argument

Function call করার সময়:

```
add(5,3);
```

এখানে:

```
5,3 = Arguments
```

Mental Trick:

```
Parameter

↓
Placeholder


Argument

↓
Actual Value
```

Example:

```
int square(int x)
{
    return x*x;
}
```

Call:

```
square(5);
```

Here:

```
x = parameter

5 = argument
```

Chapter 5 — Return Value Pattern

অনেক CP problem:

```
Calculate Something
        ↓
Return Answer
```

Example:

Maximum:

```
int maximum(int a,int b)
{
    if(a>b)
        return a;

    return b;
}
```

Usage:

```
int ans;

ans = maximum(10,20);
```

Return-এর কাজ

Function থেকে:

```
Answer পাঠিয়ে দেয়
```

back to caller.

Example:

```
main

↓

calculate()

↓

return answer

↓

main receives answer
```

Chapter 6 — Void Function Pattern

সব function value return করে না।

Example:

Print function:

```
void hello()
{
    printf("Hello");
}
```

এখানে:

```
Return নেই
```

তাই:

```
void
```

Example:

```
void printArray(int arr[], int n)
{
    for(int i=0;i<n;i++)
    {
        printf("%d ",arr[i]);
    }
}
```

Call:

```
printArray(arr,n);
```

Difference:

Return Function

```
int sum()
{
    return answer;
}
```

Output:

```
Value দেয়
```

Void Function

```
void print()
{
    printf();
}
```

Output:

```
কাজ করে
কিন্তু value দেয় না
```

Chapter 7 — Breaking Problems into Functions

এটাই Day 6-এর সবচেয়ে important part।

Problem:

> Check array-এর মধ্যে কয়টা prime number আছে।

Beginner:

সব main-এর ভিতরে:

```
Input

Loop

Prime Check

Count
```

সব একসাথে।

Professional:

Break:

```
main()

      |
      |

isPrime()

      |

countPrime()
```

Structure:

```
int isPrime(int n)
{
    // prime logic
}


int main()
{
    for(...)
    {
        if(isPrime(arr[i]))
        {
            count++;
        }
    }
}
```

Mental Model:

```
One Big Problem

        ↓

Small Independent Problems

        ↓

Functions
```

Chapter 8 — Common Function Mistakes

Mistake 1 — Return Missing

Wrong:

```
int add(int a,int b)
{
    a+b;
}
```

Problem:

Function promised:

```
int return
```

কিন্তু কিছু return করলো না।

Correct:

```
return a+b;
```

Mistake 2 — Wrong Return Type

Wrong:

```
int hello()
{
    printf("Hi");
}
```

যদি value return না করে:

```
void hello()
```

Mistake 3 — Function Before Declaration

Wrong:

```
int main()
{
    add(2,3);
}


int add(int a,int b)
{

}
```

C আগে জানতে চায় function আছে।

Solution:

Prototype:

```
int add(int,int);
```

বা function আগে লিখো।

Mistake 4 — Changing Original Data Accidentally

Array function-এ:

```
void change(int arr[])
```

Array reference হিসেবে যায়।

তাই modify করলে original change হয়।

Chapter 9 — Day 6 Assignment

Part A — Basic Functions

Create:

```
1. add()

2. subtract()

3. multiply()

4. maximum()

5. minimum()
```

Part B — Array Functions

Write:

```
1. printArray()

2. sumArray()

3. maxArray()

4. minArray()

5. countEven()
```

Part C — Pattern Conversion

এই code:

```
main()
{
   prime logic
   count logic
}
```

কে convert করো:

```
main()

isPrime()

countPrime()
```

Part D — Practice Problems

Solve:

Problem 1

Array-এর sum function দিয়ে বের করো।

Problem 2

Array-এর largest element function দিয়ে বের করো।

Problem 3

একটি number palindrome কিনা function দিয়ে check করো।

Problem 4

একটি string vowel count function দিয়ে বের করো।

Day 6 Final Checklist

```
[ ] Function কেন দরকার বুঝি

[ ] Function syntax জানি

[ ] Parameter vs Argument বুঝি

[ ] Return value বুঝি

[ ] Void function বুঝি

[ ] Function call flow বুঝি

[ ] বড় problem break করতে পারি

[ ] Array function লিখতে পারি

[ ] Prime function লিখতে পারি

[ ] Code modular করতে পারি
```

Day 6 Final Mental Model

আজকের পুরো chapter:

```
Problem
↓
Break Into Smaller Tasks
↓
Create Functions
↓
Give Input
↓
Function Does Work
↓
Return Result
↓
Combine Solution
```

Day 6 complete করলে তুমি আর শুধু code লিখবে না, code organize করতেও শুরু করবে।

পরেরটা হবে:

Day 7 — Searching + Binary Search Foundation
(এখান থেকে CP-এর আসল algorithmic thinking শুরু হবে।)