📘 DSA with C++ lecture 1&2

Lecture-1

1. History of Human Counting using Stone 🪨

- Socho aadmi jungle me rehta tha, koi paper-pen nahi, bas nature.
- Jab bhi usne **gai-bail, bheed-bakri** ginne hote the,
  woh **pathar/pebbles** alag-alag rakh leta.
- Example: 5 bakri → 5 pathar. Agar ek bakri gayab ho gayi to ek pathar hata diya.

  👉 Ye tha **first data structure** in human history → "stone collection".

---

2. Egypt Number System (Base-60, Decimal, Arithmetic) 🏺

- Egypt aur Mesopotamia ke log base 60 aur base 10 use karte the.
- Base-60 system ka fayda → clock, angle measurement (360°) aaj bhi usi ka result hai.
- Arithmetic waha kaafi practical tha: kheton ka area nikalna, anaaj ka hisaab rakhna.

---

3. Computer Evolution 💻

- 1940s → ENIAC, ek bada hall bhar ke machine hota tha.
- Usme **vacuum tubes** hote the → current flow control karte the.
- Problem → bahut heat generate hoti thi, aur machine bar-bar fail hoti thi.

---

4. Transistor Concept 🔋

- 1947 me transistor invent hua.
- Ek transistor ek **switch** hai → ON/OFF kar sakta hai.
- Ye ON/OFF later binary 0/1 ban gaya.

  👉 Isi se binary number system computer ke liye perfect fit bana.

---

5. Binary Number System (0 aur 1 ki duniya) ⚡

- Computer sirf **current hai → 1, current nahi → 0** samajhta hai.
- Human → 10 fingers se ginta hai → base 10.
- Machine → 2 state hai (on/off) → base 2.

  👉 Yehi hai binary system.

---

6. Conversion Decimal → Binary ✍️

Example: 13 (decimal) ko binary me convert karo.

- 13 ÷ 2 = 6 remainder 1
- 6 ÷ 2 = 3 remainder 0
- 3 ÷ 2 = 1 remainder 1
- 1 ÷ 2 = 0 remainder 1

  👉 Reverse likho → **1101**

---

7. Binary → Decimal

Example: 1101 (binary)

- $(1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0)$

  $= 8 + 4 + 0 + 1 =$ **13**

---

8. Octal & Hexadecimal Introduction

- Octal = base 8 → easy conversion from binary (group of 3 bits).
- Hexadecimal = base 16 → easy for humans, compact representation.

  👉 Example: 1111 binary = F hex.

---

9. Hexadecimal Detail

- 0–9 and A–F letters use hote hain.
- Example: 255 decimal = FF in hex.
- Programming me debugging aur memory address ke liye best.

---

10. How Transistor Works (Binary → Decimal) ⚙️

- Transistor circuit ek number ko represent karne ke liye ON/OFF hoti hai.
- Bahut sare transistor milke ek **register/memory** bante hain.

  👉 Example: 8 transistor = 1 byte = 0 to 255 values.

---

11. Moore's Law 📈

- Gordon Moore ne bola → har 18-24 months me transistor count double hoga.
- Matlab computer har 2 saal me fast aur cheap hote gaye.

---

12. Machine Language

- Sabse neeche → 0 aur 1 ke instruction (10110011).
- Human ke liye tough → sirf machine samajh sakti hai.

---

13. Assembly Language

- Little better → MOV A, 5 (matlab A register me 5 daal do).
- Abhi bhi human ke liye thoda boring.

---

14. High Level Language

- C, C++, Java, Python → human readable, compiler translate karta hai.
- Example: `int a = 5;`

---

15. Data with Real Life Example

- Data = raw facts.
- Example: "Marks = 95" → fact hai.
- Processing karke report card ban gaya → information.

---

16. Need of DSA

- Jaise human apne saman ko almari me systematic rakhta hai → easy access.
- Waise hi computer me **DSA chahiye** for efficiency.

👉 Without DSA → slow code, more memory waste.

---

17. Summary

- Stone counting → binary → transistor → machine → high-level language → DSA.
- Computer ek calculator se intelligent system bana.

Lecture-2📘

1. Concept of Flowchart

- Visual representation of logic.
- Symbols → rectangle (process), diamond (decision), oval (start/end), arrows (flow).

---

2. Example of Flowchart

- Human ka daily life:
  - Wake up → Check time → If late → Run → Else → Walk normally.

---

3. Flowchart Terminology

- Start/Stop: Oval
- Input/Output: Parallelogram
- Decision: Diamond
- Process: Rectangle

---

4. Flowchart – Sum of Two Numbers

- Input A, B → Process A+B → Output SUM.

---

5. Average of Two Numbers

- Input A, B → SUM=A+B → AVG=SUM/2 → Output AVG.

---

6. Find Cube of a Number

- Input N → Cube=N*N*N → Output.

7. Find Square

- Input N → Square=N*N → Output.

8. Conditional Decision Making

- If condition true → one path
- Else → another path.

9. Outside Raining Condition ☔

- Input: IsRain?
- Yes → Take Umbrella
- No → Go freely.

10. Number Even or Odd

- Input N
- If N%2==0 → Even
- Else → Odd.

11. Basics of Operator

- Arithmetic: + - * / %
- Relational: > < == !=
- Logical: && || !

12. Number Positive or Not

- If N >= 0 → Positive
- Else → Negative.

13. Loop Concept 🔄

- Repeat action multiple times without rewriting code.

14. Print a Number 10 Times

- For i=1 to 10 → print N.

15. Print n Natural Numbers

- For i=1 to N → print i.

16. Sum of n Natural Numbers

- SUM=0 → For i=1 to N → SUM+=i.

17. Prime Number Check

- Input N
- Check divisibility 2 to √N
- If divisible → Not Prime
- Else → Prime.

Conclusion

- Lecture-1 → Human history se computer aur DSA ka birth.
- Lecture-2 → Problem-solving ke liye **Flowchart aur Logic Building**.
- Next step → Algorithms aur C++ implementation.

📘 Flowchart Diagrams (Lecture-2)

4. **Sum of Two Numbers**

```
[Start]
   |
   v
```

```
[Input A,B]
   |
   v
[SUM = A + B]
   |
   v
[Output SUM]
   |
   v
 [End]
```

5. **Average of Two Numbers**

```
 [Start]
   |
   v
[Input A,B]
   |
   v
[SUM = A + B]
   |
   v
[AVG = SUM / 2]
   |
   v
[Output AVG]
   |
   v
 [End]
```

6. **Cube of a Number**

```
[Start]
   |
   v
```

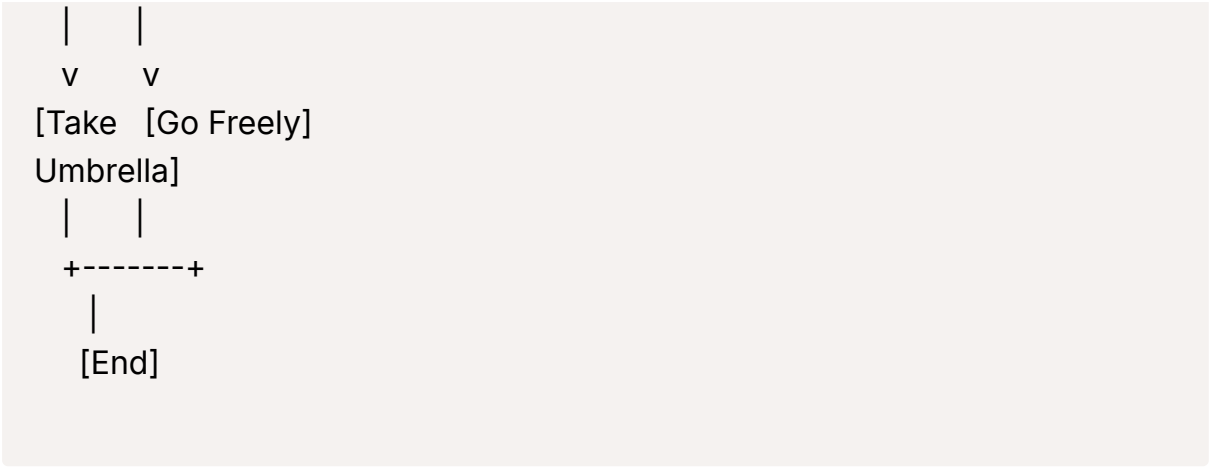
```
[Input N]
   |
   v
[CUBE = N*N*N]
   |
   v
[Output CUBE]
   |
   v
 [End]
```

7. **Square of a Number**

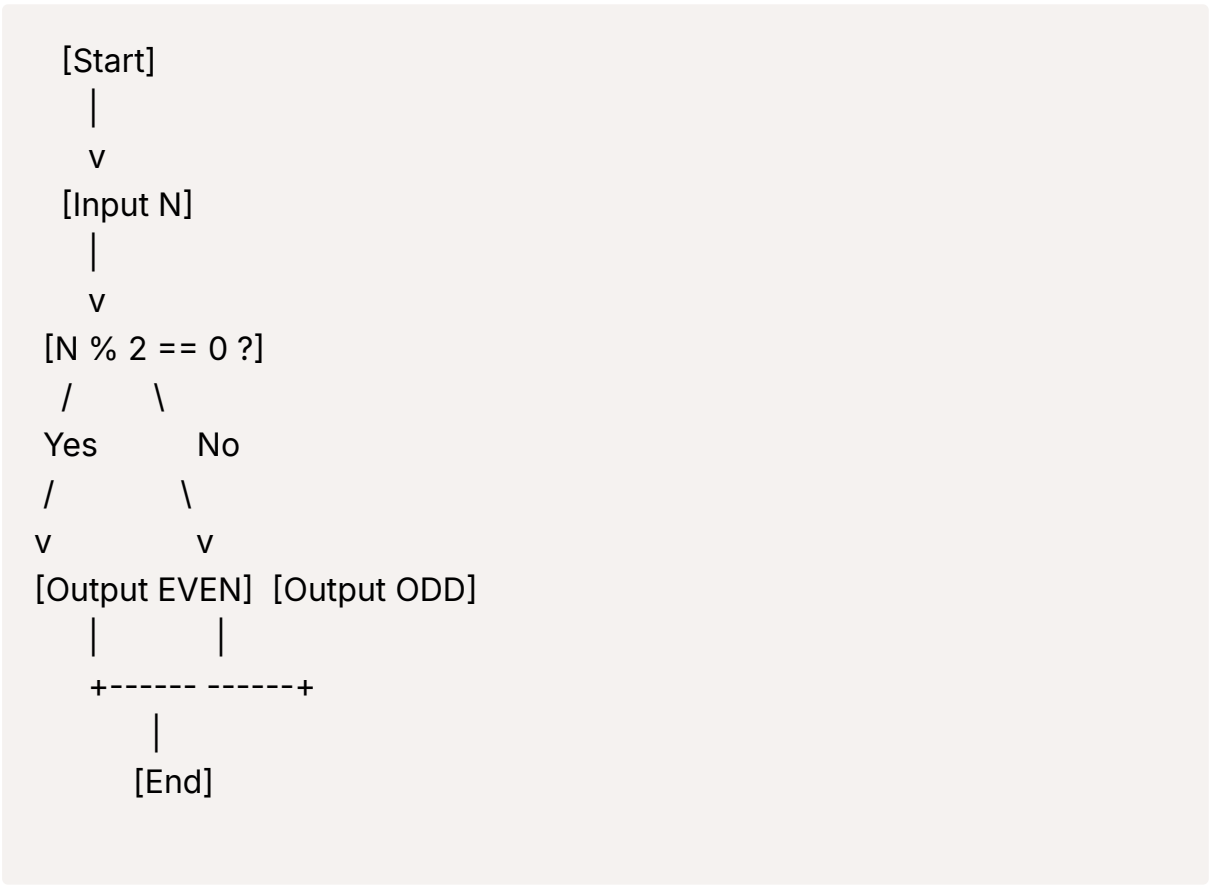
```
[Start]
   |
   v
[Input N]
   |
   v
[SQUARE = N*N]
   |
   v
[Output SQUARE]
   |
   v
 [End]
```

9. **Outside Raining Condition** ☔

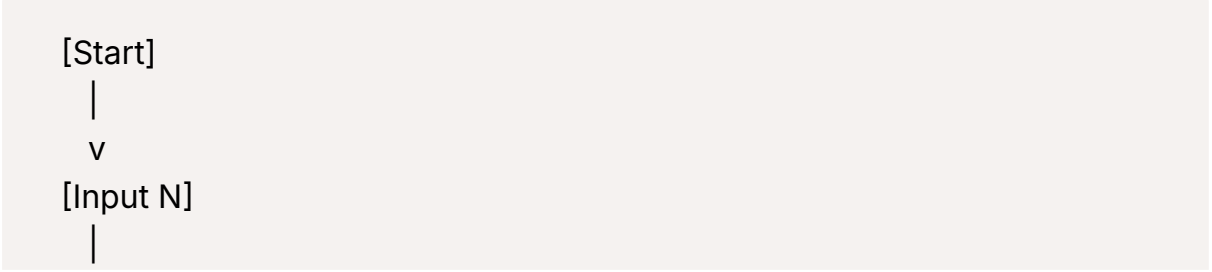
```
  [Start]
     |
     v
[Is it Raining?]
     |
  +--Yes--+
```

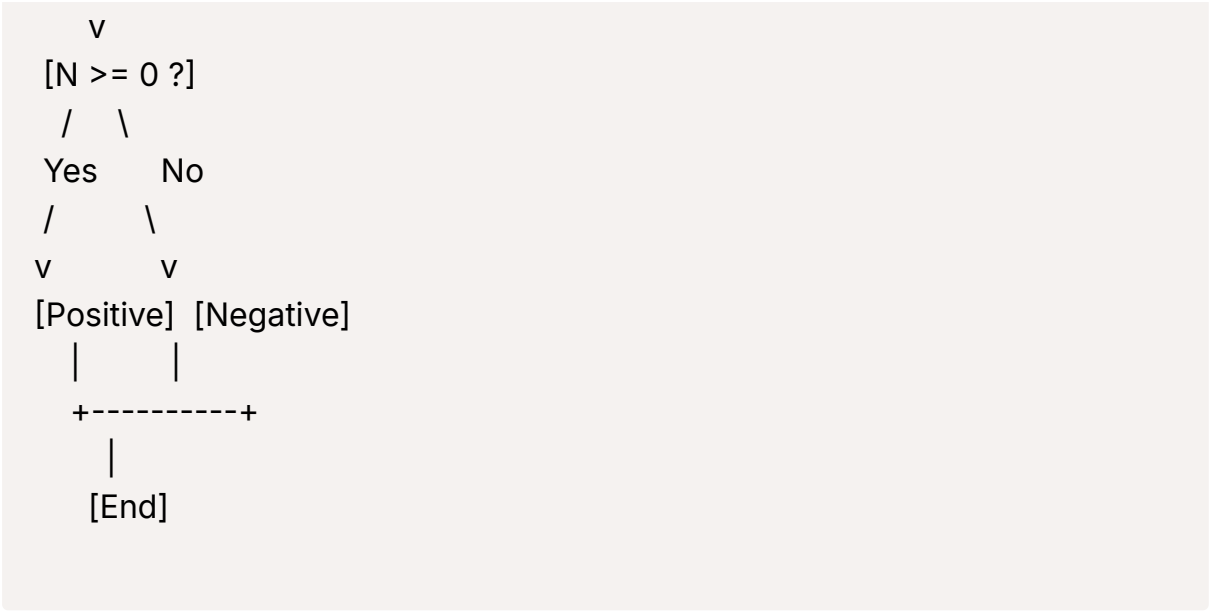


10. **Even or Odd**

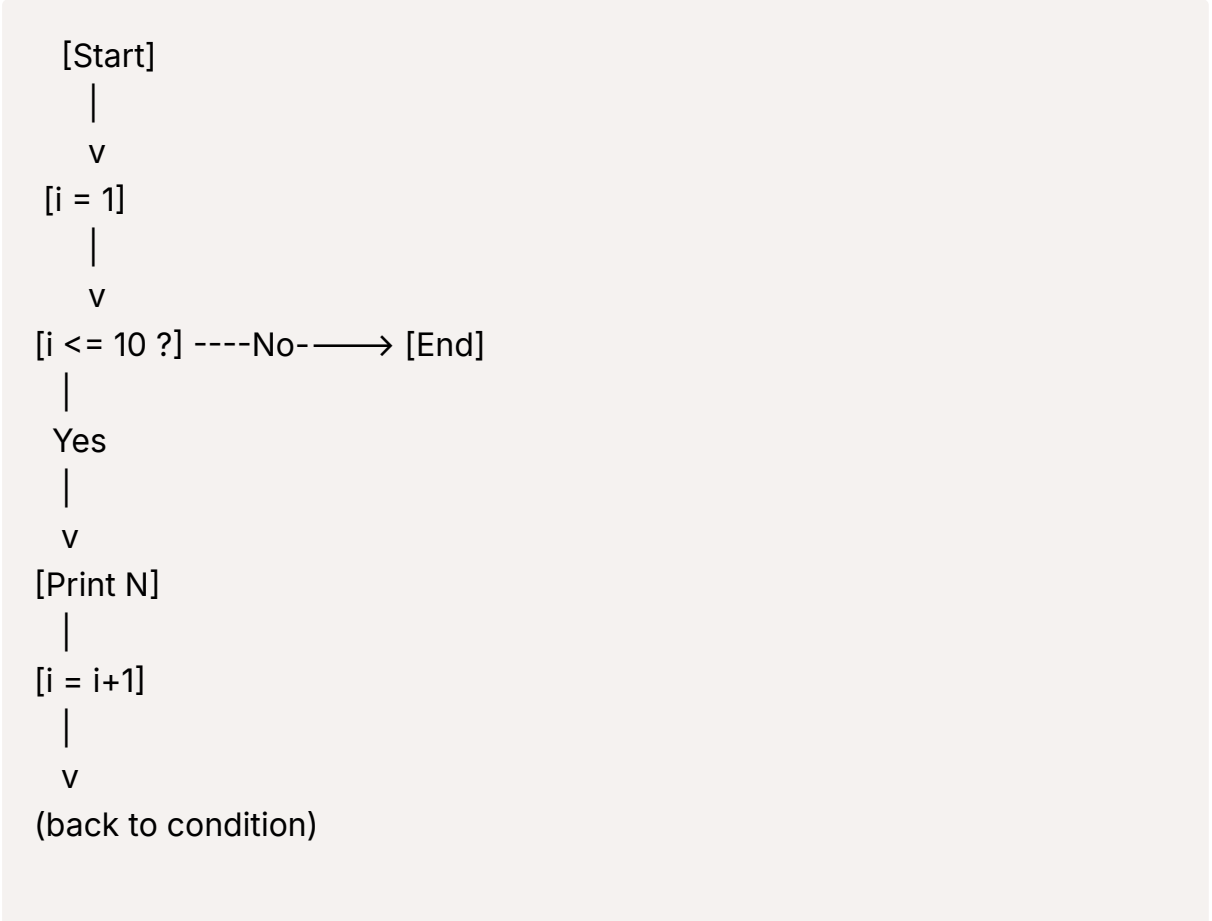


12. **Positive or Negative**

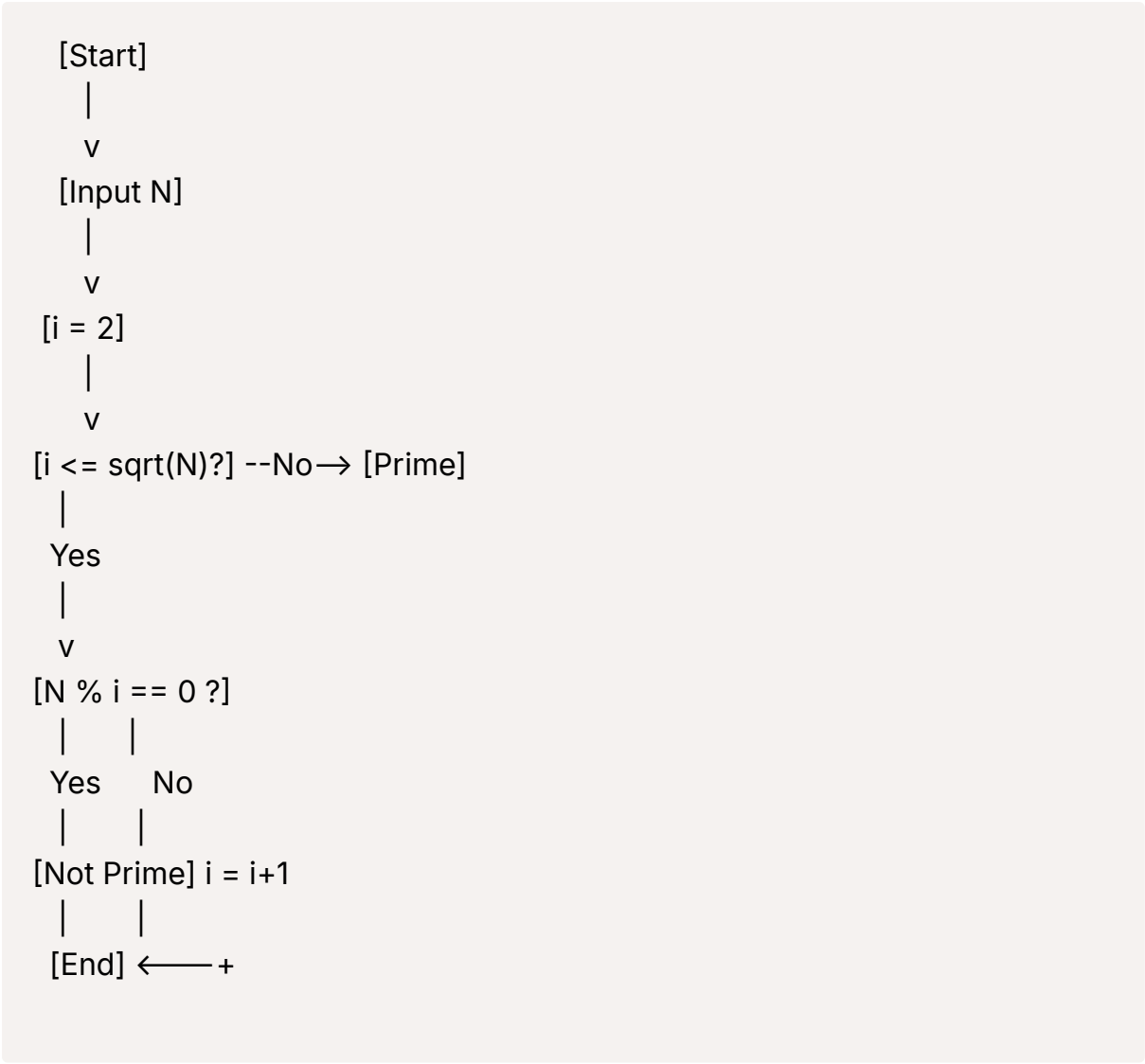




14. **Print a Number 10 Times**



17. **Prime Number Check**



---

handwritten lecture 1 & 2 notes