🌟 Number System — Complete Notes with Conversion 🌈

📌 What is a Number System?

A **Number System** is a way to represent numbers using a consistent set of **symbols** (digits). Just like humans speak **languages**, computers use **number systems** to process and store data.

💡 Analogy: Number System as Languages 🗣️

Language	Script Used
English	A-Z
Computers	0, 1, 2...F (depending on base)

➡️ Think of **Binary** as the language of computers, and **Decimal** as the language of humans.

🔢 Types of Number Systems

Number System	Base	Digits Used	Common Use
Binary	2	0, 1	Computers 💻
Octal	8	0 - 7	File permissions (Linux) 🐧
Decimal	10	0 - 9	Daily life 🔢
Hexadecimal	16	0 - 9, A - F	Memory addresses 💾

🧠 Memory Hook: BODH 🔁

Binary → **O**ctal → **D**ecimal → **H**exadecimal 🌱 Learn once, and convert between all easily!

🔄 Conversion Methods

🎯 1. **Decimal → Binary / Octal / Hexadecimal**

✅ **Method**: Repeated Division by Base

```
Step 1: Divide the number by the base
Step 2: Note the remainder
Step 3: Repeat until quotient is 0
Step 4: Read remainders bottom to top
```

Example: Decimal 25 → Binary

```
25 ÷ 2 = 12 R1
12 ÷ 2 = 6  R0
6 ÷ 2  = 3  R0
3 ÷ 2  = 1  R1
1 ÷ 2  = 0  R1
Answer = 11001
```

➡️ **Decimal 25 = Binary 11001**

🎯 2. **Binary → Decimal**

✅ **Method**: Multiply each bit by `2^position` and add

```
Binary: 1101
= (1×2^3) + (1×2^2) + (0×2^1) + (1×2^0)
= 8 + 4 + 0 + 1 = 13
```

🎯 3. **Binary → Octal**

✅ Group bits in 3s from right to left

```
Binary: 110101 → Group: 110 101 → Octal: 6 5 → Answer = 65
```

🎯 4. **Binary → Hexadecimal**

✅ Group bits in 4s from right

```
Binary: 11101110 → Group: 1110 1110 → Hex: E E → Answer = EE
```

🎯 5. **Hexadecimal → Decimal**

```
Hex: 1F
= (1×16^1) + (15×16^0)
= 16 + 15 = 31
```

🎯 Shortcuts 🎯

Binary	Hex	Octal
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	—
1001	9	—
1010	A	—
1011	B	—
1100	C	—
1101	D	—
1110	E	—
1111	F	—

💻 JavaScript Code Examples

◇ Decimal to Binary

```javascript
let dec = 25;
let binary = dec.toString(2); // "11001"
console.log(binary);
```

◇ Binary to Decimal

```javascript
let binary = "11001";
let decimal = parseInt(binary, 2); // 25
console.log(decimal);
```

🔹 Hexadecimal to Decimal

```
let hex = "1F";
let decimal = parseInt(hex, 16); // 31
console.log(decimal);
```

🔹 Decimal to Hexadecimal

```
let dec = 255;
let hex = dec.toString(16); // "ff"
console.log(hex.toUpperCase()); // "FF"
```

🧠 Bonus: Use Cases

Field	Uses Number System
Programming	Binary, Hex (colors, memory)
Networking	Binary (IP addressing)
Security/Cryptography	Hex, Octal
File Systems	Octal (Linux permissions)

📝 Summary with Emoji Cheat Sheet

System	Base	Symbol	Memory Hook
Binary ◍	2	0,1	Computers
Octal ◍	8	0-7	Linux 🐧
Decimal ◍	10	0-9	Humans 🧠
Hexadecimal ◍	16	0-9, A-F	Memory 💾

🎯 Final Tips

🔹 Always group bits from **right to left** for conversion. 🔹 Use `parseInt(str, base)` in JS for conversions.
🔹 Remember: `1 Hex Digit = 4 Bits` `1 Octal Digit = 3 Bits`

🔢 ✨ Number System Conversions Cheat Sheet 🔄

Master the art of converting between 🧮 different number systems like a pro! Let’s explore how to convert between 👉 Decimal, Binary, Octal, and Hexadecimal using examples, logic, and C++ code!

🧠 Number Systems Explained

🔢 System	⚙️ Base	🧩 Digits Used
🔟 Decimal	10	0️⃣1️⃣2️⃣3️⃣4️⃣5️⃣6️⃣7️⃣8️⃣9️⃣
💻 Binary	2	0️⃣1️⃣
🐙 Octal	8	0️⃣ to 7️⃣
🧪 Hexadecimal	16	0️⃣–9️⃣ and A-F 🅰️🅱️🆎🆑🅾️🆘

1️⃣ Decimal ➡️ Binary / Octal / Hexadecimal

🔁 Logic:

- Keep dividing by the base (2/8/16).
- Save the remainders.
- Read them in reverse 🔄.

📌 Example: Convert 25_{10} to Binary

```
25 ÷ 2 = 12 R1
12 ÷ 2 = 6  R0
6 ÷ 2  = 3  R0
3 ÷ 2  = 1  R1
1 ÷ 2  = 0  R1
Binary: 11001₂ ✅
```

📌 Convert 125_{10} to Octal

```
125 ÷ 8 = 15 R5
15 ÷ 8  = 1  R7
1 ÷ 8   = 0  R1
Octal: 175₈ ✅
```

📌 Convert 255_{10} to Hex

```
255 ÷ 16 = 15 R15 (F)
15 ÷ 16  = 0  R15 (F)
Hex: FF₁₆ ✅
```

2️⃣ Binary ➡️ Decimal / Octal / Hexadecimal

📌 Convert 1101_2 to Decimal

$(1\times2^3) + (1\times2^2) + (0\times2^1) + (1\times2^0) = 8+4+0+1 = 13_{10}$ ☑

📌 Convert 101110_2 to Octal

Group: 101 110
$101_2 = 5_8$
$110_2 = 6_8$
Octal: 56_8 ☑

📌 Convert 11011011_2 to Hex

Group: 1101 1011
$1101_2 = D$
$1011_2 = B$
Hex: DB_{16} ☑

3️⃣ Octal ➡️ Decimal / Binary

📌 Convert 157_8 to Decimal

$(1\times8^2) + (5\times8^1) + (7\times8^0) = 64+40+7 = 111_{10}$ ☑

📌 Convert 57_8 to Binary

$5_8 = 101_2$
$7_8 = 111_2$
Binary: 101111_2 ☑

4️⃣ Hexadecimal ➡️ Decimal / Binary / Octal

📌 Convert $2F_{16}$ to Decimal

$(2\times16^1) + (15\times16^0) = 32 + 15 = 47_{10}$ ☑

📌 Convert `A3₁₆` to Binary

```
A = 1010₂
3 = 0011₂
Binary: 10100011₂ ✅
```

📌 Convert `1F₁₆` to Octal

```
1 = 0001₂
F = 1111₂
Binary = 00011111
Group in 3s → 001 111 1 = 0011111₂
Octal = 37₈ ✅
```

🧾 Quick Reference Table

🔟 Decimal	💻 Binary	🐙 Octal	🧪 Hex
1	0001_2	1_8	1_{16}
2	0010_2	2_8	2_{16}
3	0011_2	3_8	3_{16}
4	0100_2	4_8	4_{16}
5	0101_2	5_8	5_{16}
6	0110_2	6_8	6_{16}
7	0111_2	7_8	7_{16}
8	1000_2	10_8	8_{16}
9	1001_2	11_8	9_{16}
10	1010_2	12_8	A_{16}
11	1011_2	13_8	B_{16}
12	1100_2	14_8	C_{16}
13	1101_2	15_8	D_{16}
14	1110_2	16_8	E_{16}
15	1111_2	17_8	F_{16}

🛠️ C++ Code for Conversions

✅ Decimal ➡️ Binary / Octal / Hex

```
#include <iostream>
#include <stack>
using namespace std;

void decimalToBase(int num, int base) {
    stack<char> st;
    string digits = "0123456789ABCDEF";
    while (num > 0) {
        st.push(digits[num % base]);
        num /= base;
    }
    while (!st.empty()) {
        cout << st.top();
        st.pop();
    }
    cout << endl;
}

int main() {
    int num;
    cout << "Enter Decimal Number: ";
    cin >> num;

    cout << "🔢 Binary: ";
    decimalToBase(num, 2);

    cout << "🐙 Octal: ";
    decimalToBase(num, 8);

    cout << "🧪 Hex: ";
    decimalToBase(num, 16);

    return 0;
}
```

▶️ Example

Input: `num = 29` **Output:**

```
Binary: 11101
Octal: 35
Hex: 1D
```

✅ Binary ➡️ Decimal / Octal / Hex

```
#include <iostream>
#include <cmath>
```

```
using namespace std;

int binaryToDecimal(long long binary) {
    int decimal = 0, i = 0;
    while (binary > 0) {
        decimal += (binary % 10) * pow(2, i++);
        binary /= 10;
    }
    return decimal;
}

int main() {
    long long binary;
    cout << "Enter Binary Number: ";
    cin >> binary;

    int decimal = binaryToDecimal(binary);
    cout << "Decimal: " << decimal << endl;
    cout << "Octal: " << oct << decimal << endl;
    cout << "Hex: " << hex << decimal << endl;

    return 0;
}
```

☑ Octal ➡ Decimal / Binary / Hex

```
#include <iostream>
#include <cmath>
#include <bitset>
using namespace std;

int octalToDecimal(int octal) {
    int decimal = 0, i = 0;
    while (octal > 0) {
        decimal += (octal % 10) * pow(8, i++);
        octal /= 10;
    }
    return decimal;
}

int main() {
    int octal;
    cout << "Enter Octal Number: ";
    cin >> octal;

    int decimal = octalToDecimal(octal);
    cout << "Decimal: " << decimal << endl;
    cout << "Binary: " << bitset<16>(decimal) << endl;
    cout << "Hex: " << hex << decimal << endl;
    return 0;
}
```

🧾 Conversions You Now Know:

✅ Decimal ↔ Binary / Octal / Hex ✅ Binary ↔ Decimal / Octal / Hex ✅ Octal ↔ Decimal / Binary ✅ Hex ↔ Decimal / Binary / Octal

---

🔚 **Congratulations!** You’ve unlocked the full power of number system conversions! 🚀 💡