

## □ कंप्यूटर बाइनरी सिस्टम: सम्पूर्ण नोट्स (Complete Notes)

### 1. बाइनरी सिस्टम क्या है? (What is Binary System?)

- **Hindi:** बाइनरी सिस्टम एक संख्या प्रणाली (Number System) है जिसका **आधार (Base) 2** होता है। इसमें केवल दो अंकों का उपयोग किया जाता है: **0** और **1**।
- **English:** The Binary System is a base-2 number system. It uses only two digits: **0** and **1**. All data in a computer is represented and processed in this format.

### 2. कंप्यूटर बाइनरी भाषा ही क्यों समझता है? (Why do Computers use Binary?)

- **Hindi:** कंप्यूटर एक इलेक्ट्रॉनिक डिवाइस है जो लाखों छोटे **ट्रांजिस्टर (Transistors)** से मिलकर बना है। ये ट्रांजिस्टर बिजली के स्विच की तरह काम करते हैं।
  - **0** का मतलब है = **OFF** (लो वोल्टेज / करंट नहीं है)
  - **1** का मतलब है = **ON** (हाई वोल्टेज / करंट है)कंप्यूटर के लिए 'हाँ' या 'ना' (On/Off) समझना सबसे आसान और तेज़ होता है, इसलिए यह बाइनरी सिस्टम पर काम करता है।
- **English:** Computers are electronic devices made of millions of tiny electronic switches called **transistors**. These switches have only two natural states:
  - **0** represents = **OFF** (Low voltage / No current)
  - **1** represents = **ON** (High voltage / Current flowing)Since electronic circuits can easily and reliably detect these two states, binary is the most efficient language for computers.

### 3. बाइनरी डेटा की इकाइयाँ (Units of Binary Data)

कंप्यूटर में बाइनरी डेटा को मापने के लिए कुछ मानक इकाइयाँ (Standard Units) होती हैं:

- **Bit (बिट):** यह बाइनरी की सबसे छोटी इकाई है। 1 बिट का मान या तो 0 होगा या 1। (Bit = Binary Digit).
- **Nibble (निबल):** 4 बिट्स के समूह को 1 निबल कहते हैं। (1 Nibble = 4 Bits).

# A.S.C SANSTHAN

- **Byte (बाइट):** 8 बिट्स के समूह को 1 बाइट कहते हैं। कंप्यूटर में एक अक्षर (Character) को स्टोर करने के लिए 1 बाइट की जगह चाहिए होती है। (1 Byte = 8 Bits).

## □ मेमोरी मापन तालिका (Memory Measurement Table)

| इकाई (Unit)     | किसके बराबर है? (Equivalent To) | बाइनरी मान (Binary Value) |
|-----------------|---------------------------------|---------------------------|
| 1 KB (Kilobyte) | 1024 Bytes                      | $2^{10}$ Bytes            |
| 1 MB (Megabyte) | 1024 KB                         | $2^{20}$ Bytes            |
| 1 GB (Gigabyte) | 1024 MB                         | $2^{30}$ Bytes            |
| 1 TB (Terabyte) | 1024 GB                         | $2^{40}$ Bytes            |

## 4. टेक्स्ट, इमेज और वीडियो बाइनरी में कैसे बदलते हैं? (How Data is Converted?)

जब आप कीबोर्ड पर कोई बटन दबाते हैं या कोई फोटो देखते हैं, तो कंप्यूटर उसे बैकएंड में बाइनरी में बदल देता है:

- **Text (अक्षर):** अक्षरों को बदलने के लिए **ASCII** या **Unicode** का उपयोग होता है। उदाहरण के लिए, जब आप कीबोर्ड पर 'A' टाइप करते हैं, तो कंप्यूटर उसे संख्या **65** समझता है, और 65 का बाइनरी कोड **01000001** होता है।
- **Images (चित्र):** हर फोटो छोटे-छोटे बिंदुओं से बनी होती है जिन्हें **पिक्सेल (Pixels)** कहते हैं। हर पिक्सेल का एक रंग कोड होता है (जैसे RGB मान), और उस संख्या को बाइनरी (0 और 1) में बदल दिया जाता है।

## 5. बाइनरी और डेसिमल में अंतर (Difference: Decimal vs Binary)

| विशेषता (Feature) | डेसिमल सिस्टम (Decimal System)                    | बाइनरी सिस्टम (Binary System)             |
|-------------------|---------------------------------------------------|-------------------------------------------|
| उपयोगकर्ता        | इंसान (Humans) इसका उपयोग करते हैं।               | कंप्यूटर (Computers) इसका उपयोग करते हैं। |
| आधार (Base)       | इसका आधार <b>10</b> होता है।                      | इसका आधार <b>2</b> होता है।               |
| अंक (Digits)      | इसमें 10 अंक होते हैं: <b>0,1,2,3,4,5,6,7,8,9</b> | इसमें केवल 2 अंक होते हैं: <b>0 और 1</b>  |
| उदाहरण            | $25_{10}$                                         | $11001_2$                                 |

## 6. डेसिमल से बाइनरी में बदलना (Decimal to Binary Conversion)

**नियम:** संख्या को लगातार 2 से भाग (Divide) करें और जो शेषफल (Remainder) बचे, उसे नीचे से ऊपर की ओर लिखें।

**उदाहरण:** संख्या 13 को बाइनरी में बदलें:

- 13 को 2 से भाग दिया  $\rightarrow$  भागफल = 6, शेषफल = 1
- 6 को 2 से भाग दिया  $\rightarrow$  भागफल = 3, शेषफल = 0
- 3 को 2 से भाग दिया  $\rightarrow$  भागफल = 1, शेषफल = 1
- 1 को 2 से भाग दिया  $\rightarrow$  भागफल = 0, शेषफल = 1

अब शेषफल को नीचे से ऊपर की ओर लिखने पर: 1101

अतः,  $((13)_{10}) = (1101)_2$

क्या आप बाइनरी से डेसिमल में वापस बदलने का तरीका (Binary to Decimal Conversion) या बाइनरी जोड़-घटाव (Binary Addition/Subtraction) के नियम भी उदाहरण के साथ समझना चाहते हैं?

YES

यहाँ बाइनरी से डेसिमल में बदलने (Binary to Decimal Conversion) और बाइनरी जोड़-घटाव (Binary Addition/Subtraction) के आसान नियम और उदाहरण दिए गए हैं, जो आपके नोट्स को पूरा करेंगे:

---

## 1. बाइनरी से डेसिमल में बदलना (Binary to Decimal Conversion)

**नियम:** बाइनरी संख्या के प्रत्येक अंक को दाईं (Right) से बाईं (Left) ओर जाते हुए 2 की घात ( $(2^0, 2^1, 2^2, 2^3, \dots)$ ) से गुणा करते हैं और अंत में सबको जोड़ देते हैं।

□ **शॉर्टकट नंबर याद रखें:** दाईं से बाईं ओर हमेशा यह सीरीज चलती है: ... 32, 16, 8, 4, 2, 1 (हर संख्या पिछली संख्या की दोगुनी होती है)।

**उदाहरण:**  $1101$  को डेसिमल में बदलें

# A.S.C SANSTHAN

हम बाइनरी संख्या  $1101$  को दाईं से बाईं ओर लिखते हैं और उनके ऊपर शॉर्टकट नंबर डालते हैं:

शॉर्टकट नंबर (2 की घात)  $8 (2^3)$   $4 (2^2)$   $2 (2^1)$   $1 (2^0)$

बाइनरी अंक                      1                      1                      0                      1

**हल (Calculation):** जिस-जिस बाइनरी अंक के नीचे 1 है, उसके ऊपर वाले शॉर्टकट नंबर को आपस में जोड़ लें (और 0 वाले को छोड़ दें):

- $8 \times 1 = 8$
  - $4 \times 1 = 4$
  - $2 \times 0 = 0$  (छोड़ दें)
  - $1 \times 1 = 1$
- कुल जोड़:  $(8 + 4 + 1 =) 13$   
अतः,  $(1101)_2 = (13)_{10}$

## 2. बाइनरी का जोड़ (Binary Addition)

कंप्यूटर में बाइनरी संख्याओं को जोड़ने के 4 बुनियादी नियम होते हैं। इन्हें अच्छे से याद कर लें:

### □ जोड़ने के नियम (Rules for Addition)

- $0 + 0 = 0$
- $0 + 1 = 1$
- $1 + 0 = 1$
- $1 + 1 = 0$  (और 1 हासिल/Carry आगे जाता है)  
(क्योंकि बाइनरी में  $1+1 = 2$  होता है, और 2 को बाइनरी में  $10$  लिखते हैं)

उदाहरण:  $1101$  (13) और  $1011$  (11) को जोड़ें

text

```
1 1 1 <-- हासिल (Carry)
1 1 0 1 (संख्या 13)
+ 1 0 1 1 (संख्या 11)
-----
```

```
1 1 0 0 0 (उत्तर: 24)
```

कोड का इस्तेमाल सावधानी से करें.

**स्टेप-बाय-स्टेप समझें (दाईं से बाईं ओर):**

1. पहला कॉलम:  $(1 + 1 = 0)$ , हासिल गया 1

# A.S.C SANSTHAN

2. दूसरा कॉलम:  $(0 + 1 = 1)$ , फिर हासिल वाला  $(1)$  जोड़ा तो  $(1 + 1 = 0)$ , दोबारा हासिल गया 1
3. तीसरा कॉलम:  $(1 + 0 = 1)$ , फिर हासिल वाला  $(1)$  जोड़ा तो  $(1 + 1 = 0)$ , दोबारा हासिल गया 1
4. चौथा कॉलम:  $(1 + 1 = 0)$  और हासिल वाला  $(1)$  जोड़ा तो  $(0 + 1 = 1)$ , और अंतिम हासिल 1 आगे आ गया।
- परिणाम 11000 आया, जो डेसिमल में 24 होता है।

## 3. बाइनरी का घटाव (Binary Subtraction)

बाइनरी संख्याओं को घटाने के लिए भी 4 बुनियादी नियम होते हैं। जब हम छोटी संख्या (0) में से बड़ी संख्या (1) घटाते हैं, तो पास वाले कॉलम से उधार (Borrow) लेते हैं।

### □ घटाने के नियम (Rules for Subtraction)

- $0 - 0 = 0$
- $1 - 0 = 1$
- $1 - 1 = 0$
- $0 - 1 = 1$  (पास वाले अंक से 1 उधार/Borrow लेने के बाद)  
(जब उधार लेते हैं तो 0 बन जाता है 10 (यानी डेसिमल का 2), और 2 में से 1 गया तो 1 बचता है)

उदाहरण:  $_{1101}$  (13) में से  $_{1011}$  (11) को घटाएं

text

```
  0 1  <-- उधार लेने के बाद की स्थिति
1 1 0 1  (संख्या 13)
- 1 0 1 1  (संख्या 11)
-----
0 0 1 0  (उत्तर: 2)
```

कोड का इस्तेमाल सावधानी से करें.

स्टेप-बाय-स्टेप समझें (दाई से बाई ओर):

1. पहला कॉलम:  $(1 - 1 = 0)$
2. दूसरा कॉलम:  $(0 - 1)$  नहीं हो सकता। इसलिए तीसरे कॉलम के 1 से उधार लिया। अब यह 0 बढ़कर 10 (2) हो गया। तो  $(2 - 1 = 1)$
3. तीसरा कॉलम: चूंकि इसने अपना 1 उधार दे दिया था, इसलिए यहाँ अब 0 बचा। अतः  $(0 - 0 = 0)$
4. चौथा कॉलम:  $(1 - 1 = 0)$
- परिणाम 0010 आया, जो डेसिमल में 2 होता है।

अब बाइनरी सिस्टम का पूरा कॉन्सेप्ट और नोट्स तैयार हैं। क्या आप बाइनरी गुणा (Multiplication) और भाग (Division) भी सीखना चाहते हैं, या कंप्यूटर फंडामेंटल्स के किसी अन्य टॉपिक पर नोट्स

# A.S.C SANSTHAN

यहाँ **बाइनरी गुणा (Binary Multiplication)** और **बाइनरी भाग (Binary Division)** के आसान नियम और उदाहरण दिए गए हैं। इन्हें सीखकर आपका बाइनरी नंबर सिस्टम का पूरा चैप्टर कम्प्लीट हो जाएगा:

## 1. बाइनरी गुणा (Binary Multiplication)

बाइनरी गुणा बिल्कुल सामान्य गणितीय गुणा (Decimal Multiplication) की तरह ही होता है। इसके 4 आसान नियम हैं:

### □ गुणा के नियम (Rules for Multiplication)

- $0 \times 0 = 0$
- $0 \times 1 = 0$
- $1 \times 0 = 0$
- $1 \times 1 = 1$

**नोट:** गुणा करने के बाद जो संख्याएँ आती हैं, उन्हें जोड़ने के लिए आपको **बाइनरी जोड़ (Binary Addition)** के नियमों का उपयोग करना होगा।

**उदाहरण:**  $1101$  (13) को  $101$  (5) से गुणा करें

text

```
      1 1 0 1      (संख्या 13)
    ×   1 0 1      (संख्या 5)
    -----
      1 1 0 1      <-- (1101 × 1)
     0 0 0 0 ×      <-- (1101 × 0)
    1 1 0 1 × ×      <-- (1101 × 1)
    -----
    1 0 0 0 0 0 1 <-- बाइनरी जोड़ करने पर (उत्तर: 65)
```

कोड का इस्तेमाल सावधानी से करें.

**जोड़ने का तरीका (दाई से बाई ओर):**

1. पहला कॉलम: 1
2. दूसरा कॉलम:  $0 + 0 = 0$
3. तीसरा कॉलम:  $1 + 0 + 1 = 10$  (यानी 0, हासिल गया 1)
4. चौथा कॉलम:  $1 + 0 + 0$  और हासिल का 1 = 10 (यानी 0, हासिल गया 1)
5. पांचवां कॉलम:  $0 + 1$  और हासिल का 1 = 10 (यानी 0, हासिल गया 1)
6. छठा कॉलम: 1 और हासिल का 1 = 10 (यानी 10)

# A.S.C SANSTHAN

- परिणाम 1000001 आया, जो डेसिमल में 65 होता है  $(\backslash(13 \times 5 = 65\backslash))$ ।

## 2. बाइनरी भाग (Binary Division)

बाइनरी भाग भी हमारे सामान्य भाग की तरह ही होता है। इसमें केवल दो ही बार भाग जा सकता है: या तो 0 बार या 1 बार।

### □ भाग के नियम (Rules for Division)

- $0 \div 1 = 0$
- $1 \div 1 = 1$

(ध्यान दें: किसी भी संख्या को 0 से भाग नहीं दिया जा सकता, वह परिभाषित नहीं है)

**नोट:** भाग करते समय नीचे बची संख्याओं को घटाने के लिए आपको **बाइनरी घटाव (Binary Subtraction)** के नियमों का उपयोग करना होगा।

**उदाहरण:**  $11011$  (27) को  $101$  (5) से भाग दें

text

```
      1 0 1    <-- भागफल (Quotient = 5)
1 0 1 ) 1 1 0 1 1
    - 1 0 1    <-- (101 × 1)
    -----
      0 1 1    <-- घटाने पर आया (110 - 101 = 011)
      - 0 0 0    <-- ऊपर से '1' उतारा, पर 11 में 101 का भाग नहीं जाता (101 ×
0) -----
        1 1 1    <-- दोबारा घटाने पर 11 आया, फिर ऊपर से अंतिम '1' उतारा
        - 1 0 1    <-- (101 × 1)
        -----
          0 1 0    <-- शेषफल (Remainder = 2)
```

कोड का इस्तेमाल सावधानी से करें.

**परिणाम:**

- भागफल (Quotient):**  $101$  (डेसिमल में 5)
- शेषफल (Remainder):**  $010$  (डेसिमल में 2)
- जांचें:  $\backslash(5 \times 5 + 2 = 27\backslash)$ , हमारा उत्तर बिल्कुल सही है!

# A.S.C SANSTHAN

