✳ CODEARMY DSA

✳ LECTURE - 1

- Earlier people used stones to count eg → a person used stones to count no of goats

First type of counting was tally marks.

Egyp used to use Base 60 number system.
But India brought decimal no. system — PROUD 

Decimal : {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
↳ Base 10

→ This no. system was easier to do arithmetic operations.

Earlier arithmetic operations were difficult to calculate manually.

Then came computer but that were used to be big as of size of a room.

Then came transistors → devic to amplify electrical signals that have two output working or not working.

working - 1
not working - 0

Notes

✳ Binary no system

$$\Rightarrow \begin{Bmatrix} 0 \\ 1 \end{Bmatrix} = \text{Base}_2$$

21

Tuesday

January 2020

Week 04

Day 021 - 345

Date 21 - 01 - 2020

JAN 20	M	T	W	T	F	S	S
			1	2	3	4	5
	6	7	8	9	10	11	12
	13	14	15	16	17	18	19
	20	21	22	23	24	25	26
	27	28	29	30	31		

9.00

General additions

0	0	1	1	10	11
+0	+1	+0	+1	+1	+1
0	1	1	10	11	100

10.00

* Converting Decimal to Binary

11.00

Eg → 27 ⇨ keep on diving by 2

12.00

2	27	1
2	13	1
1.00 2	6	0
2	3	1
	1	

2.00

⇒ 11011 → Ans

3.00 ⇒ 47

2	47	1
2	23	1
4.00 2	11	1
2	5	1
5.00 2	2	0
	1	

⇒ 101111 → Ans

6.00

7.00 ⇒ Converting Binary to Decimal

eg → 11011

$(1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

Notes

16 + 8 + 0 + 2 + 1 ⇒ 27

You can't change the direction of the wind but can adjust the sail.

January 2020

Week 04

Day 022 - 344

Date 22 - 01 - 2020

22

Wednesday

⇒ PRACTICE

1.) 37

2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
	1	

⇒ 100101

2.) 92

2	92	0
2	46	0
2	23	1
2	11	1
2	5	1
2	2	0
	1	

⇒ 1011100

3.) 128

2	128	0
2	64	0
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
	1	

⇒ 10000000

1.) 1011

$(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$8 \quad 0 \quad 2 \quad 1 \Rightarrow 11$

2.) 111001

$(1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

$32 \quad 16 \quad 8 \quad 0 \quad 0 \quad 1$

⇒ 57

Notes 3.) 10011011

128 0 (32) 16 8 0

⇒ $(1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$2 \quad 1$

Success is a journey, not a destination. ⇒ 155

23
Thursday

January 2020
Week 04
Day 023 - 343
Date 23 - 01 - 2020

JAN 20

M	T	W	T	F	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31		

9.00 ✳ OCTAL NUMBER SYSTEM
⇒ {0, 1, 2, 3, 4, 5, 6, 7}

10.00 → Converting to octal from decimal eg → 23

8	23	7
8	2	2
	0	

⇒ 27

11.00

→ Converting octal to decimal

12.00 ⇒ $(2 \times 8^1) + (7 \times 8^0)$

1.00 ↓

$16 + 7 \Rightarrow 23$

2.00 ✳ HEXADECIMAL NUMBER SYSTEM
⇒ {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

3.00

⇒ Converting decimal to hexadecimal

4.00 eg → 11

16	11	
	0	B

5.00 B → Ans

6.00 → here we use (A, B, C, D, E, F) because if A would have been 10 then we would have interpreted as

$(1 \times 16^1) + (0 \times 16^0) \rightarrow 16 + 1 = 17$

that would have given wrong answer.

7.00

Notes

→ Converting Hexadecimal to decimal

⇒ AC2 2560

⇒ $(10 \times 16^2) + (12 \times 16^1) + (2 \times 16^0)$

↓ 192 ↓ 2

⇒ 2754

⇒ Transistors are used to store data as data first gets converted to binary and then we have set of transistors that follow binary pattern of data. By this we can store various type of data.

eg → how 13 will be stored

13 → 1101

T1	- OFF
T2	- OFF
T3	- OFF
T4	- OFF
T5	- OFF
T6	- OFF
T7	- ON
T8	- ON
T9	- OFF
T10	- ON

→ So this is how transistors by getting ON/OFF stores data in binary as

T-OFF = 0 | T-ON = 1

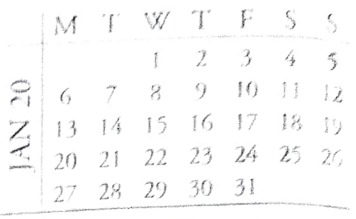
T3	T2	T3	
0	0	0	= 0
0	0	1	= 1
0	1	0	= 2
0	1	1	= 3
1	0	0	= 4
1	0	1	= 5
1	1	0	= 6
1	1	1	= 7

So set of 3 transistors can store 8 different no. so transistors become very efficient.

Moore Law: Every 2 years no of transistors in IC doubles as transistor size decreases.

Notes

→ Language of computer is called Machine Language (Binary) but we people understant English Language & Maths much better



So we code in high level language that gets converted to binary.

But earlier programmers used to communicate with computers using binary language only that was too tough. Then came Assembler that made programming. Programmers used to code in Assembly language & then a software assembler used to convert it to binary & reverse but to make it much simpler

→ Came high level language that used English to write code that made coding very easy. to interprut/ convert it to binary was done by software compiler / interpreter.

Machine Language	Assembly Language	High Level Language
-	-	
Fast	slow	- much slower

⟹ In today's world we have huge data that must be stored very efficiently such that it must take less space & to fetch data it must take very less time.

Notes

End goal → arrange data efficiently

→ Some must be with code

* Calculating sum upto n no.

→ So our code must be such that it must take least time to calculate answer.

Code 1

1 + 2 + 3 . . . n

⇒ more time consuming

Code 2

use formula $\frac{n*(n+1)}{2}$

⇒ much faster

* Lecture 1 H.W Solutions

- Convert Decimal to binary

1.)

2	37	1
2	18	0
2	9	1
2	4	0
2	2	0
	1	

→ 100101

2.)

2	92	0
2	46	0
2	23	1
2	11	1
2	5	1
2	2	0
	1	

→ 1011100

3.) 128

2	128	0
2	64	0
2	32	0
2	16	0
2	8	0
2	4	0
2	2	0
	1	

→ 10000000

4 → 243

2	243	1
2	121	1
2	60	0
2	30	0
2	15	1
2	7	1
2	3	1

→ 11110011

Notes

- Convert Binary to decimal

1.) 1011

→ $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$

$\downarrow 8 \quad \downarrow 0 \quad \downarrow 2 \quad \downarrow 1$

⇒ 11

2.) 111001

→ $(1 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

⇒ 57

3.) 10011011

→ $(1 \times 2^7) + (0 \times 2^6) + (0 \times 2^5) + (1 \times 2^4) + (1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) + (1 \times 2^0)$ ⇒ 155

4.) 10100100

⇒ $(1 \times 2^7) + (0 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0)$ ⇒ 128 + 32 + 4

⇒ 164

- Decimal to octal

1.) 28

8	28	4
	3	3
	0	

⇒ 34

2.) 47

8	47	7
	5	5
	0	

→ 57

Notes

Everything that irritates us about others can lead us to an understanding of ourselves.

January 2020

Week 05

Day 028 - 338

Date 28 - 01 - 2020

28

Tuesday

3.) 9280

8	9280	0
8	116	4
8	14	6
	1	

⇒ 1640

4.) 1243

8	1243	3
8	155	3
8	19	3
	2	

→ 2333

- octal to decimal

1.) 41 → $4\times8^1 + 1\times8^0$ → 33

2.) 207 → $2\times8^2 + 0\times8^1 + 7\times8^0$ → 128 + 7 → 135

3.) 124 → $1\times8^2 + 2\times8^1 + 4\times8^0$ → 64 + 16 + 4 → 84

4.) 311 → $3\times8^2 + 1\times8^1 + 1\times8^0$ → 192 + 8 + 1 → 201

- Decimal to Hexadecimal

1.) 317

16	317	D
16	19	3
	1	

⇒ 13D

2.) 41

16	41	9
16	2	2
	0	

→ 29

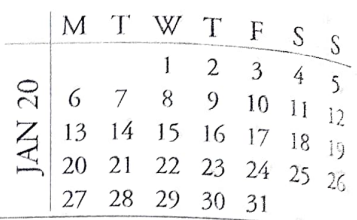
3) 14 → 16 | 14 | E

0

→ E

Notes

Good judgment comes from experience and experience comes from poor judgment.



4.) 845

16	845	D
16	52	4
	3	

⇒ 34D

- Hexadecimal to decimal

1.) A11

→ $(10 \times 16^2) + (1 \times 16^1) + (1 \times 16^0)$ ⇒ 2560 + 16 + 1

⇒ 2577

2.) 49

⇒ $(4 \times 16^1) + (9 \times 16^0)$ ⇒ 64 + 9 ⇒ 73

3.) AE2F

→ $(10 \times 16^3) + (14 \times 16^2) + (2 \times 16^1) + (15 \times 16^0)$

40960 + 3584 + 32 + 15

⇒ 44591

4.) D97

→ $(13 \times 16^2) + (9 \times 16^1) + (7 \times 16^0)$

⇒ 3328 + 144 + 7

⇒ 3479

Notes