Week02—Activity 02

1. What are different encoding schemes. Discuss in details.

ASCII

ASCII is an American standard as designed to encode English characters and punctuation as used on typewriters and teletypes of that era (1960s).

ASCII uses 8 bits although only 7 bits are actually used.

Because ASCII was developed at the time Teletype devices were in operation it also contains **control codes** designed to control the teletype device.

ASCII Table

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	0	0		32	20	40	[space]	64	40	100	@	96	60	140	`
1	1	1		33	21	41	!	65	41	101	A	97	61	141	a
2	2	2		34	22	42	"	66	42	102	B	98	62	142	b
3	3	3		35	23	43	#	67	43	103	C	99	63	143	c
4	4	4		36	24	44	$	68	44	104	D	100	64	144	d
5	5	5		37	25	45	%	69	45	105	E	101	65	145	e
6	6	6		38	26	46	&	70	46	106	F	102	66	146	f
7	7	7		39	27	47	'	71	47	107	G	103	67	147	g
8	8	10		40	28	50	(	72	48	110	H	104	68	150	h
9	9	11		41	29	51	)	73	49	111	I	105	69	151	i
10	A	12		42	2A	52	*	74	4A	112	J	106	6A	152	j
11	B	13		43	2B	53	+	75	4B	113	K	107	6B	153	k
12	C	14		44	2C	54	,	76	4C	114	L	108	6C	154	l
13	D	15		45	2D	55	-	77	4D	115	M	109	6D	155	m
14	E	16		46	2E	56	.	78	4E	116	N	110	6E	156	n
15	F	17		47	2F	57	/	79	4F	117	O	111	6F	157	o
16	10	20		48	30	60	0	80	50	120	P	112	70	160	p
17	11	21		49	31	61	1	81	51	121	Q	113	71	161	q
18	12	22		50	32	62	2	82	52	122	R	114	72	162	r
19	13	23		51	33	63	3	83	53	123	S	115	73	163	s
20	14	24		52	34	64	4	84	54	124	T	116	74	164	t
21	15	25		53	35	65	5	85	55	125	U	117	75	165	u
22	16	26		54	36	66	6	86	56	126	V	118	76	166	v
23	17	27		55	37	67	7	87	57	127	W	119	77	167	w
24	18	30		56	38	70	8	88	58	130	X	120	78	170	x
25	19	31		57	39	71	9	89	59	131	Y	121	79	171	y
26	1A	32		58	3A	72	:	90	5A	132	Z	122	7A	172	z
27	1B	33		59	3B	73	;	91	5B	133	[	123	7B	173	{
28	1C	34		60	3C	74	<	92	5C	134	\	124	7C	174	|
29	1D	35		61	3D	75	=	93	5D	135	]	125	7D	175	}
30	1E	36		62	3E	76	>	94	5E	136	^	126	7E	176	~
31	1F	37		63	3F	77	?	95	5F	137	_	127	7F	177	

Unicode

Because of the need to encode foreign language symbols and other graphic characters the Unicode character set and encoding schemes were developed.

The most common encoding schemes are:

- UTF-8
- UTF-16
- UTF-32

UTF-8 is the most commonly used encoding scheme used on today's computer systems and computer networks.

It is a variable width encoding scheme and was designed to be fully backwards compatible with ASCII. It uses 1 to 4 bytes.

Character Sets and Encoding Schemes

The distinction between the two isn't always clear and the terms tend to be used interchangeable.

A character set is a list of characters whereas an encoding scheme is how they are represented in binary.

This is best seen with Unicode.

The encoding schemes UTF-8, UTF-16 and UTF-32 use the Unicode character set but encode the characters differently.

ASCII is a character set and encoding scheme.

2. **Convert the binary number 11001 to decimal**

3. **Covert the binary number 11010010 to a decimal number**

   210

4. **Convert the decimal number 45 to binary**

   101101

5. **Convert the hexadecimal number B2 to binary**

   10110010

6. **Convert the binary number 11011 to hexadecimal**

   1B

7. **Convert the decimal number 20 to hexadecimal**

   14

8. **Convert the hexadecimal number 2C to decimal**

   44

9. **Convert the binary number 10101100 to its decimal equivalent**

   172

10. **Convert the decimal number 168 to binary equivalent**

    10101000

11. **Convert the hexadecimal number 0x2301 to its binary equivalent**

    10001100000001

How to convert floating numbers into binary numbers ?

Let’s take an example

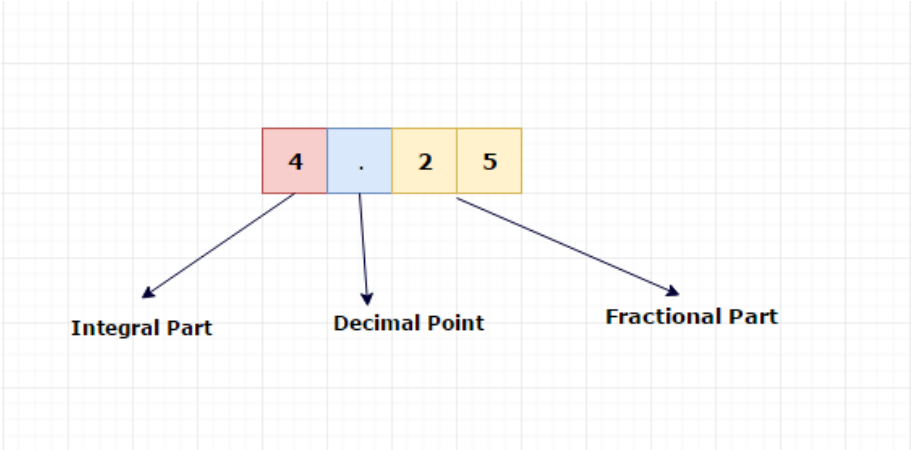
4.25

Where,

4 is an integral part.

0.25 is a fractional part.

Pictorial Explanation



Integral Part (4)

To convert an integral part into binary, just follow the binary number system method.

Using that method, we can represent 4 as (100) 2.

Fractional part (0.25)

To convert the fractional part to binary, multiply fractional part with 2 and take the one bit which appears before the decimal point.

Follow the same procedure with after the decimal point (.) part until it becomes 1.0.

Like,

0.25 * 2 =0.50 //take 0 and move 0.50 to next step

0.50 * 2 =1.00 //take 1 and stop the process

0.25 = (01) 2

Combining both integral and fractional,

4.25 = (100.01) 2