Chapter 10

Number Systems and Arithmetic Operations

10.1 The Decimal Number System:

The Decimal number system is a number system of base or radix equal to 10, which means that there are 10, called Arabic numerals, symbols used to represent number : 0, 1, 2, 3,…….,9 , which are used for counting.

To represent more than nine units, we must either develop additional symbols or use those we have in combination. When used in combination, the value of the symbol depends on its position in the position in the combination of symbols. We refer to this as positional notation and refer to the position as having a weight designated as units, tens, hundreds, thousands, and so on.

The units symbol occupies the first position to the left of the decimal point is represented as 10^0. The second position is represented as 10^1, and so forth. To determine what the actual number is in each position, take the number that appears in the position, and multiply it by 10^X, where x is the power representation.

This is expressed mathematically of the first five positions as

10^4	10^3	10^2	10^1	10^0
Ten thousands	thousands	hundreds	tens	units

For example the value of the combination of symbols 435 is determined by adding the weight of each position as

$$4 \times 10^2 + 3 \times 10^1 + 5 \times 10^o$$

Which can be written as

$$4 \times 100 + 3 \times 10 + 5 \times 1$$

Or $400 + 30 + 5 = 435$

The position to the right of the decimal point carry a positional notation and corresponding weight as well. The exponents to the right of the decimal point are negative and increase in integer steps starting with-1. This is expressed mathematically for each of the first four positions as;

weight	10^{-1}	$1\ 0^{-2}$	10^{-3}	10^{-4}
	tenths	hundredths	thousandths	ten thousandths

For example the value of the combination of symbols, 249.34 determined by adding the weight of each position as

$$2 \times 10^2 + 4 \times 10^1 + 9 \times 10^\circ + 3 \times 10^{-1} + 4 \times 10^{-2}$$

Or $$200 + 40 + 9 + \frac{3}{10} + \frac{4}{100}$$

Or $$200 + 40 + 9 + 0.3 + 0.04$$

$$= 249.34$$

10.2 The Binary Number System:

The binary number system is a number system of base or radix equal to 2, which means that there are two symbols used to represent number : 0 and 1.

A seventeenth-century German Mathematician, Gottfriend Wilhelm Von Leibniz, was a strong advocate of the binary number system. The binary number system has become extremely important in the computer age.

The symbols of the binary number system are used to represent number in the same way as in the decimal system symbol is used individually; then the symbols are use combination. Since there are only two symbols, we can represent two numbers , 0 and 1, with individual symbols. The position of the 1 or 0 in a binary number system indicates its weight or value within the number. We then combine the 1 with 0 and with itself to obtain additional numbers.

10.3 Binary and Decimal Number Correspondence:

Here are first 15 equivalence decimal and binary numbers:

Decimal Number	**Binary Numbers**
0	0000
1	0001
2	0010
3	0011

4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

An easy way to remember that how to write a binary sequence such as in the above table for a four-bits example is as follows:

1. The right most column in the binary number begins with a 0 and alternate each bit.
2. The next column begin with two O's and alternate every two bits.
3. The next column begin with four 0's and alternate every four bits.
4. The next column begin with eight-O's and alternate every eight bits.

It is seen that it takes at least four bits from 0 to 15. The formula to Count the decimal number with n bits, beginning with zero is:

Highest decimal number = $2^n - 1$

for example, with two bits we can count the decimal number from 0 to 3 as,

$2^2 - 1 = 3$

For three bits, the decimal number is from 0 to 7, as,

$$2^3 - 1 = 7$$

The same type of positional weighted system is used with binary numbers as in the decimal system, The base 2 is raised to power equal to the number of positions away from the binary point The weight and designation of the several positions are as follows:

Power equal to position Base

weight	4	3	2	1	0	-1	-2	-3
positional notation	2	2	2	2	2	2	2	2
(decimal value)	16	8	4	2	1	0.5	0.25	0.125

when the symbols 0 and 1 are used to represent binary number, each symbol is called a binary digit or a bit. Thus the binary number 1010 is a four-digit binary number or a 4-bit binary number,

10.4 Binary-to-Decimal Conversion:

Since we are programmed to count in the decimal number system, it is only natural that we think in terms of the decimal equivalent value when we see a binary number. The conversion process is straight forward and is done as follows: Multiply binary digit (1 or 0) in each position by the weight of the position and add the results. The following examples explain the process.

Example 1: Convert the following binary number to their decimal equivalent. (a) 1101 (b) 1001

Solution:

(a) 1101 $= (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^o)$

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

(b) 1001 $= (1 \times 2^3) + (0 \times 2^2) + (0 \times 2^1)(1 \times 2^0)$

$= 8 + 0 + 0 + 1 = 9$

Example 2: Convert the following binary numbers to their decimal equivalent. (a) 0.011 (b) 0.111

Solution:

(a) 0.011 $= (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$

$= 0 + \frac{1}{4} + \frac{1}{8}$

$= 0.25 + 0.125 = 0.375$

(b) 0.111 $= (1 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$

$= \frac{1}{2} + \frac{1}{4} + \frac{1}{8}$

$= 0.5 + 0.25 + 0.125 = 0.875$

Example 3: Convert the binary number 110.011 to its decimal equivalent.

Solution:

$$110.011 = (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^\circ) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$$

$$= 4 + 2 + 0 + 0 + \frac{1}{4} + \frac{1}{8}$$

$$= 4 + 2 + 0.25 + 0.125 = 6.375$$

10.5 Decimal-to-Binary Conversion:

It is frequently necessary to convert decimal numbers to equivalent binary numbers. The two most frequently used methods for making the conversion are the

- Repeated division-by-2 or multiplication-by-2 method.

Which is discussed below:

10.6 Repeated Division-by-2 Or Multiplication-by-2 Method:

To convert a decimal whole number to an equivalent number in a new base, the decimal number is repeatedly divided by the new base. For the case of interest here, the new base is 2, hence the repeated division by 2. Repeated division by 2 means that the original number is divided by 2, the resulting quotient is divides by 2, and each resulting quotient thereafter is divided by 2 until the quotient is 0. The remainder resulting from each division forms binary number. The first remainder to be produced is called the least significant bit (LSB) and the last remainder is called most significant bit (MSB).

When converting decimal fraction to binary, multiply repeatedly by 2 any fractional part. The equivalent binary number is formed from the

1 or 0 in the units position. The following examples illustrate the procedure.

Example 4: Convert the decimal number 17 to binary.

Solution:

2	17	
2	8 – 1 →	L.S.B.
2	4 – 0	
2	2 – 0	
2	1 – 0	
	0 – 1 →	M.S.B.

Therefore, 17 =10001

Example 5: Convert the decimal number 0.625 to binary.

Solution:

$0.625 \times 2 = 1.250$ 1 (M S B)

$0.250 \times 2 = 0.500$ 0

$0.25 \times 2 = 1.00$ 1 (L S B)

Therefore, 0.625 = 0.101

Note: Any further multiplication by 2 in example 5 will equal to 0; therefore the multiplication can be terminated. However, this, is not so. Often it will be necessary to terminate the multiplication when an acceptable degree of accuracy is obtained. The binary number obtained will then be an approximation.

Example 6: Convert the number 0.6 to binary:

Solution: Carry

$0.6 \times 2 = 1.2$	1 (M S B)
$0.2 \times 2 = 0.4$	0
$0.4 \times 2 = 0.8$	0
$0.8 \times 2 = 1.6$	1
$0.6 \times 2 = 1.2$	1 (LSB)

Therefore, $0.6 = 0.10011$

10.7 Double-Dibble Technique:

To convert a binary integer to a decimal integer we make use of double-dibble technique. The verb dibble is a neologism (i.e., a made-up-word) which has found wide spread acceptance among programmer's and other computer-oriented persons. To dibble a number is to double it and then add 1. The double-dibble technique for converting a binary integr (whole-number) goes a follows:

Begin by setting the first 'results equal to 1. If the second digit of the binary number is a zero then double this 1 (= 2) and if the second binary digit is a 1, then dibble this 1 (= 3) to obtain the second result; continue to double or dibble the successive results according to whether the successive binary digits are 0 or 1; the result corresponding to the last binary digit is the decimal equivalent of the binary integer.

Example 7: Convert 110101101 to a decimal number.

Solution:

Binary digits		1	1	0	1	0	1	1	0	1
Results	double	1		6		26			214	
	double		3		13		53	107		429

Thus 110101101 = 429

10.8 The Octal Number System:

The octal number system is used extensively in digital work because it is easy to convert from octal to binary, vice versa. The octal system has a base; or radix, of 8, which means that there are eight symbols which are used to form octal numbers. Therefore, the single-digit numbers of the octal, number system are

0 1, 2, 3, 4, 5, 6, 7

To count beyond 7, a 1 is carried to the next higher-order column and combined with each of the other symbols, as in the decimal system. The weight of the different positions far the octal system is the base raised to the appropriate power, as shown below

weight	3	2	1	0	-1	-2
positional notation	8	8	8	8	8	8
(decimal value)	512	64	8	1	$\frac{1}{8}$	$\frac{1}{64}$

Octal numbers look just like decimal numbers except that the symbols 8 and 9 are not used. To distinguish between octal and decimal numbers, we must subscript the numbers with their bas. for example, $20_8 = 16_{10}$.

The following table shows octal numbers 0 through 37 and their decimal equivalent.

Octal numbers and their Decimal equivalent

Octal	Decimal	Octal	Decimal	Octal	Decimal	Octal	Decimal
0	0	10	8	20	16	30	24
1	1	11	9	21	17	31	25
2	2	12	10	22	18	32	26
3	3	13	11	23	19	33	27
4	4	14	12	24	20	34	28
5	5	15	13	25	21	35	29
6	6	16	14	26	22	36	30
7	7	17	15	27	23	37	31

10.9 Octal-to-Decimal Conversion:

Octal numbers are converted to their decimal equivalent by multiplying the weight of each position by the digit in that position and adding the products. This is illustrated in the following examples.

Example 8: **Convert the following octal numbers to their decimal, equivalent.**

Solution: (a) 35_8 (b) 100_8 (c) 0.24_8

(a) $35_8 = (3 \times 8^1) + (5 \times 8^o)$

$= 24 + 5 = 29$

(a) $100_8 = (1 \times 8^2) + (0 \times 8^1) + (0 \times 8^o)$

$= 64 + 0 + 0 = 64_{10}$

(b) $0.24_8 = (2 \times 8^{-1}) + (4 \times 8^{-2})$

$= \frac{2}{8} + \frac{4}{64}$

$= 0.3125_{10}$

10.10 Decimal-to-Octal Conversion:

To convert decimal numbers to their octal equivalent, the following procedures are employed:

- Whole-number conversion: Repeated division-by-8.
- Fractional number conversion: Repeated multipli-cation-by-8.

10.11 Repeated Division-by-8 Method:

The repeated-division by 8 method of converting decimal to octal applies only to whole numbers. The procedure is illustrated in the following example.

Example 8: **Convert the following decimal numbers to their octal equivalent:**

(a) 245 (b) 175

Solution:

8	245	
8	30 – 5	(LSD)
8	3 – 6	
	0 – 6 (MSD)	

8	175
8	21 – 7
8	2 – 5
	0 – 2

Therefore, $245_{10} = 365_8$, therefore , $175_{10} = 257_8$

10.12 Repeated Multiplication-by-8 Method:

To convert decimal fractions to their Octal equivalent requires repeated Multiplication by 8, as shown in the following example.

Example 9: Convert the decimal fraction 0.432 to octal equivalent .

Solution:

		Carry
0.432×8	$= 3.456$	3(MSD)
0.456×8	$= 3.648$	3
0.648×8	5.184	5
0.184×8	$= 1.472$	1 (LSD)

The first carry is nearest the octal point, therefore,

$$0.432_{10} = 0.3351_8$$

The conversion to octal is not precise, since there is a remainder. If greater accuracy is required, we simply continue multiplying by 8 to obtained more octal digits.

10.13 Octal-to-Binary-Conversion:

The primary reason for our interest in octal numbers lies in entering and outputting computer data and because of the ease of octal-to-binary conversion. Computers recognize only binary information and may be programmed using only, 1's and O's.

Converting numbers from octal to binary can be done essentially by inspection. Since there are eight symbols used for counting in the octal system and eight combinations of three binary digits that corresponds to these single-digit octal numbers, we can assign a binary three-digit combination to each single-digit octal number as shown in the Table given below:

10.14 Octal and Binary Number Correspondence.

Octal	Binary
0	0 0 0
1	1 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

To convert from octal to binary, simply replace each octal digit with the corresponding thee-digit binary number, as illustrated in the following example.

Example 10: Convert the following octal numbers to their binary equivalent.

(a) 247_8 (b) 501_8

Solution:

(a)	2	4	7	Octal
	010	100	111	binary

Thus, $247_8 = 010100111_2$

(b)	5	0	1	octal
	101	000	001	binary

Thus, $501_8 = 101000001_2$

10.15 Binary-to-Octal Conversion:

In printing out octal numbers, the modern electronic digital computer performs a binary-to-octal conversion. This is a simple procedure. The binary number is divided into groups to three bits, counting to the right and to the left from the binary point and then each group Of three is interpreted as an octal digit; as shown in above table.

Example 11: Convert 11010101 . 01101 to an octal-number.

Solution:	011	010	101	.	011	10
	011	010	101		011	010
	3	2	5		3	2

Therefore $11010101.0110_1 = 325.32_8$

10.16 Binary Arithmetic:

Binary arithmetic includes the basic arithmetic operations of addition, subtraction, multiplication and division. The following sections present the rules that apply to these operations when they are performed on binary numbers.

Binary Addition:

Binary addition is performed in the same way as addition in the decimal-system and is, in fact, much easier to master. Binary addition obeys the following four basic rules:

$$\begin{array}{r} 0 \\ +\,0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\,1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\,1 \\ \hline 10 \end{array}$$

The results of the last rule may seem some what strange, remember that these are binary numbers. Put into words, the last rule states that

binary one + binary one = binary two = binary "one zero"

When adding more than single-digit binary number, carry into, higher order columns as is done when adding decimal numbers. For example 11 and 10 are added as follows:

$$\begin{array}{r} 11 \\ +\,10 \\ \hline 101 \end{array}$$

In the first column (L S C or 2°) '1 plus 0 equal 1. In the second column (2^1) 1 plus 1 equals 0 with a carry of 1 into the third column (2^2).

When we add 1 + 1.+ 1 (carry) produces 11, recorded as 1 with a carry to the next column.

Example 12: Add (a) 111 and 101 (b) 1010, 1001 and 1101.

Solution:

(a)
$$\begin{array}{r} {\scriptstyle (1)(1)} \\ 1\,1\,1 \\ 1\,0\,1 \\ \hline 1\,1\,0\,0 \end{array}$$

(b)
$$\begin{array}{r} {\scriptstyle (2)(1)(1)(1)} \\ 1\;0\;1\;0 \\ 1\;0\;0\;1 \\ 1\;1\;0\;1 \\ \hline 1\,0\,0\,0\,0 \end{array}$$

Binary Subtraction:

Binary subtraction is just as simple as addition subtraction of one bit from another obey the following four basic rules

0 – 0 = 0

1 – 1 =0

1 – 0 = 1

10 – 1 = 1 with a transfer (borrow) of 1.

When doing subtracting, it is sometimes necessary to borrow from the next higher-order column. The only it will be necessary to borrow is when we try to subtract a 1 from a 0. In this case a 1 is borrowed from the next higher-order column, which leaves a 0 in that column and creates a

10 i.e., 2 in the column being subtracted. The following examples illustrate binary subtraction.

Example 13: Perform the following subtractions.

(a) 11 - 01 , (b) 11-10 (c) 100 - 011

Solution:

(a) $\begin{array}{r} 11 \\ -\ 01 \\ \hline 10 \end{array}$ (b) $\begin{array}{r} 11 \\ -\ 10 \\ \hline 01 \end{array}$ (c) $\begin{array}{r} 100 \\ -\ 011 \\ \hline 001 \end{array}$

Part (c) involves to borrows, which handled as follows. Since a 1 is to be subtracted from a 0 in the first column, a borrow is required from the next higher-order column. However, it also contains a 0; therefore, the second column must borrow the 1 in the third column. This leaves a 0 in the third column and place a 10 in the second column. Borrowing a 1 from 10 leaves a 1 in the second column and places a 10 i.e, 2 in the first column:

When subtracting a larger number from a smaller number, the results will be negative. To perform this subtraction, one must subtract the smaller number from the larger and prefix the results with the sign of the larger number.

Example 14: Perform the following subtraction 101 – 111.

Solution:

Subtract the smaller number from the larger.

$$\begin{array}{r} 111 \\ -\ 101 \\ \hline 010 \end{array}$$

Thus 1 0 1 – 1 1 1 = - 010 = - 10

Binary multiplication:

Binary multiplication is performed in the same manner as decimal multiplication. It is much easier, since there are only two possible results of multiplying two bits. The Binary multiplication obeys the four basic rules.

$0 \times 0 = 0$

$0 \times 1 = 0$

$1 \times 0 = 0$

$1 \times 1 = 1$

Example 15: Multiply the following binary numbers.

(a) 101 × 1 1 **(b) 1101 × 10**

(c) 1010 × l 0 1 **(d) 1 0 1 l × 1 0 1 0**

Solution:

(a)
$$\begin{array}{r} 101 \\ \underline{1\,1}\times \\ 101 \\ \underline{101} \\ 1111 \end{array}$$

(b)
$$\begin{array}{r} 11101 \\ \underline{1\,0}\times \\ 0000 \\ \underline{1101} \\ 11010 \end{array}$$

(c)
$$\begin{array}{r} 1010 \\ \underline{101}\times \\ 1010 \\ 0000 \\ \underline{1010} \\ \underline{110010} \end{array}$$

(d)
$$\begin{array}{r} 1011 \\ \underline{1010}\times \\ 0000 \\ 1011\times \\ 0000\times \\ \underline{1011\times} \\ 1101110 \end{array}$$

Multiplication of fractional number is performed in the same way as with fractional numbers in the decimal numbers.

Example 16: Perform the binary multiplication 0.01 × 11.
Solution:

$$\begin{array}{r} 0.01 \\ \underline{1\,1}\times \\ 01 \\ \underline{01\times} \\ 0.11 \end{array}$$

Binary Division:

Division in the binary number system employees the same procedure as division in the decimal system, as will be seen in the following examples.

Example 17: Perform the following binary division.

(a) $110 \div 11$ (b) $1100 \div 11$

Solution:

(a)	10 11) 110 11 0 0 0 0 0 0	(b)	100 11) 11000 11 0 0 0 0 0 0 0 0 0 0

Binary division problems with remainders are also treated the same as in the decimal system, as illustrates the following example.

Example 18: Perform the following binary division:

(a) 1111 ÷ 110 (b) 1100 ÷ 101

Solution:

(a)	10. 1 110) 1 11 1.00 1 10 1 10 1 10 0 0 0	(b)	10. 011 110) 1 100.00 1 01 100 000 1000 101 110 101 1 (reminder)

EXERCISE 10

Q.1: Convert the following binary numbers to decimal equivalent.

(a) 100 (b) 11010 (c) 10110010 (d) 1.001
(e) 110100.010011 (f) 11010.10110 (g) 1000001.111

Q.2: Convert the following decimal numbers to binary equivalent.

(a) 16 (b) 247 (c) 962.84 (d) 0.0132
(e) 6.74

Q.3: Convert the following octal numbers to their decimal equivalent:

(a) 14_8 (b) 236_8 (c) 1432_8 (d) 0.43_8

(e) 0.254_8 (f) $(16742.3)_8$ (g) $(206.104)_8$

Q.4: Convert the following decimals numbers to their octal equivalent:

(a) 29 (b) 68 (c) 125 (d) 243.67
(e) 419.95 (f) 645.7 (g) 39.4475 (h) 49.21875

Q.5: Convert the following octal numbers to their binary equivalent:

(a) 13_8 (b) 27_8 (c) 65_8 (d) 124.375_8
(e) 217.436_8

Q.6: Convert the following binary numbers to their octal equivalent:

(a) 10110010 (b) 10101101

(c) 1110101 (d) 10111.101

(e) 111010.001

Q.7: Convert 18 × 24 to binary form and then perform binary multiplication.

Q.8: Convert 58.75 ÷ 23.5 to binary form and then perform operation.

Q.9: Add the following binary numbers:

(a)
```
  1 1
+ 1 1
…. …..
```

(b)
```
  1011
+ 1101
……….
```

(c)
```
  101011
+ 110101
………...
```

(d)
```
  1110101
+ 1011111
…………
```

Q.10: Add the following binary numbers:

(a)
```
  1011
+ 1101
  1011
.......
```

(b)
```
  1010110
  1110101
+ 1001010

.............
```

(c)
```
  1010110
  1111011
+ 1011111
 ..........
```

(d)
```
  10110110
  11010101
+ 11010110
 ...........
```

Q.11: Subtract the following binary numbers:

(a)
```
  1 1
– 1 0
.... .....
```

(b)
```
  111
– 101

..........
```

(c)
```
  101011
– 100101
 ..............
```

(d)
```
  11100001
– 10011110
 ............
```

Q.12: Multiply the following binary numbers:

(a) 11×11 (b) 101×10 (c) 110×101

(d) 1011×1101 (e) 1010×101

Q.13: Divide the following binary numbers:

(a) $110 \div 10$ (b) $10110 \div 10$

(d) $101101 \div 1.1$ (e) $11001.11 \div 1101$

Answers 10

Q.1: (a) 4 (b) 26 (c) 178 (d) 1.125

(e) 52.2969 (f) 26.6875 (g) 65.875

Q.2: (a) 10000_2 (b) 11110111_2 (c) 1111000010.11010111_2

(d) 0.000000110110_2 (e) 110.1011110_2

Q.3: (a) 12 (b) 158 (c) 794 (d) 0.5469

	(e)	0.3359	(f)	7650.375		(g)	134.1328		
Q.4:	(a)	35_8	(b)	104_8	(c)	175_8	(d)	363.527_8	
	(e)	643.2631_8		(f)	$(1205.043)_8$		(g)	47.345_8	
	(h)	61.61_8							
Q.5:	(a)	001011		(b)	010111		(c)	110101	
	(d)	001010100.011111101				(e) 010001111.100011110			
Q.6:	(a)	262_8	(b) 255_8	(c)	165_8	(d)	27.5_8	(e)	72.1_8
Q.7:	110110000_2								
Q.8:	10.1_2								
Q.9:	(a)	110	(b)	11000	(c)	1100000		(d) 11010100	
Q.10:	(a)	100011		(b)	100010101		(c)	100110000	
	(d)	1001100001							
Q.11:	(a)	01	(b)	10	(c)	110	(d)	1000011	
Q.12:	(a)	1001	(b)	1100	(c)	11110	(d)	110010	
	(e)	10001111							
Q.13:	(a)	11	(b)	1011	(c)	01.111111		(d)	11110

Short Questions

Write the short answers of the following:

Q.1: Define Binary Number.

Q.2: Define Octal numbers.

Q.3: Define Decimal number.

Q.4: Convert binary number 10101_2 to decimal numbers.

Q.5: Convert binary numbers 11111_2 to decimal numbers.

Q.6: Convert 110011.11_2 to decimal numbers.

Q.7: Add the binary number $110_2 + 1011_2$

Q.8: Add binary numbers $10011.1_2 + 11011.01_2$

Q.9: Multiply the binary numbers $111_2 \times 101_2$

Q.10: Subtract the binary numbers $1100_2 - 1001_2$

Q.11: Divide the binary numbers $1000_2 \div 100$

Q.12: Convert binary 101101_2 to octal nos.

Q.13: Convert binary numbers 10110111_2 to octal numbers.

Q.14: Convert binary numbers 11 0 11 0 . 0 11 to octal numbers.

Q.15: Convert octal numbers 103.45_8 to binary number.

Q.16: Convert octal number 107_8 binary nos.

Q.17: Find the octal equivalent to $(359.325)_{10}$.

Q.18: Convert the decimal numbers 932_{10} to octal numbers.

Note: LB means least bit and GB means greatest bit

Answers

Q4	21	**Q5.**	31	**Q6.**	51.75
Q7.	11000_2	**Q8.**	101110.11_2	**Q9.**	100011_2
Q10	0011_2	**Q11.**	11_2	**Q12.**	55_8
Q13.	267_8	**Q14.**	66.3_8	**Q15.**	$001000\ 0\ 11.100\ \ 101_2$
Q16.	$001\ \ 000\ \ 111_2$	**Q17.**	$(547\ .\ 246)_8$	**Q18.**	1644_8

Objective Type Questions

Q.1 Each questions has four possible answers. Choose the correct answer and encircle it.

__1. $(11)_2$ to decimal is:

(a) 3 (b) 5 (c) 4 (d) None of these

__2. $(11)_2$ to decimal is:

(a) $\frac{3}{2}$ (b) 2.2 (c) 3 (d) None of these

__3. 25 when converted to octal is:

(a) $(31)_8$ (b) $(2.5)_8$ (c) $(13)_8$ (d) None of these

__4. $(11)_2 + (101)_2$ is equal to:

(a) $(1100)_2$ (b) $(121)_2$ (c) $(112)_2$ (d) None of these

__5. Addition of $(45)_8$ and $(73)_8$ is:

(a) $(140)_8$ (b) $(118)_8$ (c) $(110)_8$

__6. $(11)_8 \times (7)_8$ is equal to:

(a) $(105)_8$ (b) $(77)_8$ (c) $(43)_8$ (d) None of these

__7. Number of digits in a binary system are:

(a) 2 (b) 7 (c) 10 (d) None of these

__8. $(11)_2 \times (11)_2$ is equal to:

(a) $(1101)_2$ (b) $(1001)_2$ (c) $(121)_2$ (d) None of these

__9. $(11)_8$ to decimal is equal to:

(a) 9 (b) 88 (c) 110 (d) None of these

__10. Conversion of 9 to binary system is:

(a) $(1001)_2$ (b) $(101)_2$ (c) $(11)_2$ (d) None of these

ANSWERS:

1.	a	2.	a	3.	d	4.	a	5.	a
6.	b	7.	a	8.	a	9.	a	10.	a