1 Define Dictionary Coding.

=> Dictionary Coding is a data compression method that replacing frequently occuring sequence of symbols with shorter codes.

This is use to reduse the overall size of the data for perforne compression.

There are Two Types of Dictionary Coding.

(i) Static Coding
(ii) Adaptive Coding.

=> Explain Static Coding with example.

Static Coding is also known as Diagram Encoding.

In Static Coding, Dictionary is Fixed for encoding and decoding.

Dictionary is Fixed for sender and receiver for compression and decompression.

Ex. abcabd

For Encode, This String we have to declare the dictionary for every character.

a - 000
b - 001
c - 010
d - 011
e - 100
f - 101
g - 110
ab - 111

→ Encoding - 000001010111011

SMVS
Page No.
Date : / /

2 Explain LZ77 encoding and decoding with example.

=> LZ77 Encoding is a type of Adaptive Coding Method.

In LZ77, we have to use Search Buffer and Look-Ahead Buffer for Encoding and decoding.

Size of Search Buffer = 7-bit
Size of Look-Ahead Buffer = 5-bit

After Encoding, we get Output in the form of (Offset, Max_match length, New_Symbol)

Here, Offset = Match character distance or pointer in Search Buffer.

Max_match_length = Length of match character in Search Buffer

New_Symbol = Next symbol of Look-Ahead Buffer.

=> Encoding Algorithm :

1 Set a sliding window or Search Buffer and a look-ahead buffer.

Add the character in look-ahead buffer.

2 Add the One by One character in Search Buffer

if match is found in Search Buffer -> Output (Offset, length, New-Symbol)

else, add into Search Buffer Output (Offset, length, New-Symbol)

3 Repeat the process until the entire input is processed.

SMVS
Page No.
Date : / /

=> Decoding Algorithm :

1 Set up Empty Buffer.

2 For each pair (offset, length) do:

IF it is literal Symbol, append to the output Buffer.

IF it is Pair, Copy the Sub-string from output buffer with offset and length.

3 Repeat the process until entire encoded data is processed.

=> ~~Exa~~

=> Example : WEDWEWEEEWEWET

	Search Buffer							Look-ahead					Output
	7	6	5	4	3	2	1	1	2	3	4	5	
								W	E	D	W	E	(0, 0, W)
							W	E	D	W	E	W	(0, 0, E)
						W	E	D	W	E	W	E	(0, 0, D)
					W	E	D	W	E	W	E	E	(3, 2, W)
			W	E	D	W	E	W	E	E	W	E	(2, 2, E)
	W	E	D	W	E	W	E	E	W	E	W	E	(3, 3, W)
WED	W	E	W	E	E	W	E	W	E	T			(2, 2, T)
													C

=> Decoding : (0, 0, W), (0, 0, E), (0, 0, D), (3, 2, W), (2, 2, E), (3, 3, W), (2, 2, T)

Input	7	6	5	4	3	2	1	
(0, 0, W)							W	
(0, 0, E)						W	E	
(0, 0, D)					W	E	D	
(3, 2, W)			W	E	D	W	E	
(2, 2, E)	W	E	D	W	E	W	E	WEDWE
(3, 3, W)	W	E	W	E	E	W	E	
(2, 2, T)	W	E	E	W	E	W	E	T

String : WEDWEWEEEWEWET

SMVS
Page No.
Date : / /

3 LZ78 Encoding and Decoding with Example.

=> LZ78 Encoding Algorithm:

1 Set the empty dictionary,
P = empty

2 While:
C = Next Character in input

If (String P+C is in the Dictionary)
than P = P+C
else,
Output the codeword to P
Output C
add the String P+C to the dictionary
P = emty.

3 If (P is not empty)
output the codeword corresponding to P

END

=> LZ78 Decoding Algorithm:

1 Set the empty dictionary

2 While (More codewords in code stream):
    W = Codeword
    C = Character
    output the string of W+C
    add the W+C to the dictionary.

end.

=> Example: WEDWEWEEWEWET

Step	Position	Output	Dictionary.
1	1	(0, W)	W
2	2	(0, E)	E
3	3	(0, D)	D
4	4	(1, E)	WE
5	5	(4, E)	WEE
6	9	(4, W)	WEW
7	13	(2, T)	ET

SMVS
Page No.
Date : / /

=> Decoding Example : <0, W> <0, E>, <0, D>, <0, E>, <4, E>, <4, W>, <2, T>

Input	Step	Pos.	Output	Dictionary
(0, W)	1	1	W	W
(0, E)	2	2	E	E
(0, D)	3	3	D	D
(1, E)	4	4	WE	WE
(4, E)	5	5	WEE	WEE
(4, W)	6	9	WEW	WEW
(2, T)	7	13	ET	ET

4 Use LZ7W Method to encode "WEDWEWEEWEWET" String.

Encoder	Output	String Codeword	String
1	W	4	WE
2	E	5	WEE
3	D	6	WEW
4	WE	7	ET
5	WEE		
6	WEW		
7	ET		