

## PRACTICAL-6

**AIM: Write a program to Implement LZ77 algorithm.**

**Code:**

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
public class LZW {
    public static void main(String[] args) {
        try
        {
            if (args.length == 2) {
                File file = new File(args[1]);
                if (file.exists()) {
                    if (args[0].toLowerCase().contains("c")) { // Compress
                        String input = Files.readString(file.toPath());
                        byte[] compressedTags = compress(input);
                        for (byte tag : compressedTags) {
                            System.out.println("Tag<" + Byte.toUnsignedInt(tag) + ">");
                        }
                        String newPath = file.getPath() + ".lzw";
                        File compressedFile = new File(newPath);
                        compressedFile.createNewFile();
                        Files.write(compressedFile.toPath(), compressedTags);
                    } else if (args[0].toLowerCase().contains("d")) { // Decompress
                        byte[] input = Files.readAllBytes(file.toPath());
                        String decompressedTxt = decompress(input);
                        System.out.println(decompressedTxt);
                        String newPath = file.getPath() + ".txt";
                        File decompressedFile = new File(newPath);
                        decompressedFile.createNewFile();
                        List<String> lines = new ArrayList<>();
                        lines.add(decompressedTxt);
                        Files.write(decompressedFile.toPath(), lines);
                    } else {
                        System.out.println(args[0] + " is invalid argument");
                    }
                } else {
                    System.out.println(args[1] + " is not an existing file");
                }
            } else {
                System.out.println("No arguments were supplied");
            }
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
    private static byte[] compress(String str) {
        ArrayList<String> dictionary = new ArrayList<>();
        List<Byte> tags = new ArrayList<>();
        for (int i = 0; i < str.length(); i++) {
            if (dictionary.size() == 128) {
                dictionary.clear();
            }
        }
    }
}
```



```

int dictionaryIndex = -1;
StringBuilder temp_str = new StringBuilder();
temp_str.append(str.charAt(i));
for (int j = i + 1; j < str.length(); j++){
    temp_str.append(str.charAt(j));
    if (dictionary.contains(temp_str.toString())) {
        dictionaryIndex = dictionary.indexOf(temp_str.toString());
        if (j == str.length() - 1) {
            byte tag = (byte) (dictionaryIndex + 128);
            tags.add(tag);
            i += temp_str.length();
        }
    }
    else {
        dictionary.add(temp_str.toString());
        byte tag = (byte) (dictionaryIndex + 128);
        if (dictionaryIndex == -1) {
            tag = (byte) str.charAt(i);
        }
        tags.add(tag);
        i += temp_str.length() - 1;
        break;
    }
}
if(str.length() - i == 1) {
    tags.add((byte) str.charAt(i));
    break;
}
byte[] bytes = new byte[tags.size()];
for (int i = 0; i < tags.size(); i++) {
    bytes[i] = tags.get(i);
}
return bytes;
}

private static String decompress(byte[] input) {
    StringBuilder decompressed = new StringBuilder();
    ArrayList<String> dictionary = new ArrayList<>();
    StringBuilder nextDic = new StringBuilder();
    nextDic.append((char) input[0]);
    decompressed.append((char) input[0]);
    for (int i = 1; i < input.length; i++) {
        if (dictionary.size() == 128) {
            dictionary.clear();
        }
        int currentTag = Byte.toUnsignedInt(input[i]) - 128;
        if(currentTag < 0) {
            nextDic.append((char) input[i]);
            decompressed.append((char) input[i]);
            dictionary.add(nextDic.toString());
            nextDic = new StringBuilder();
            nextDic.append((char) input[i]);
        }
        else {
            if(currentTag >= dictionary.size()) {
                nextDic.append(nextDic.substring(0, 1));
                decompressed.append(nextDic);
            }
            else {
                nextDic.append(dictionary.get(currentTag).charAt(0));
            }
        }
    }
}

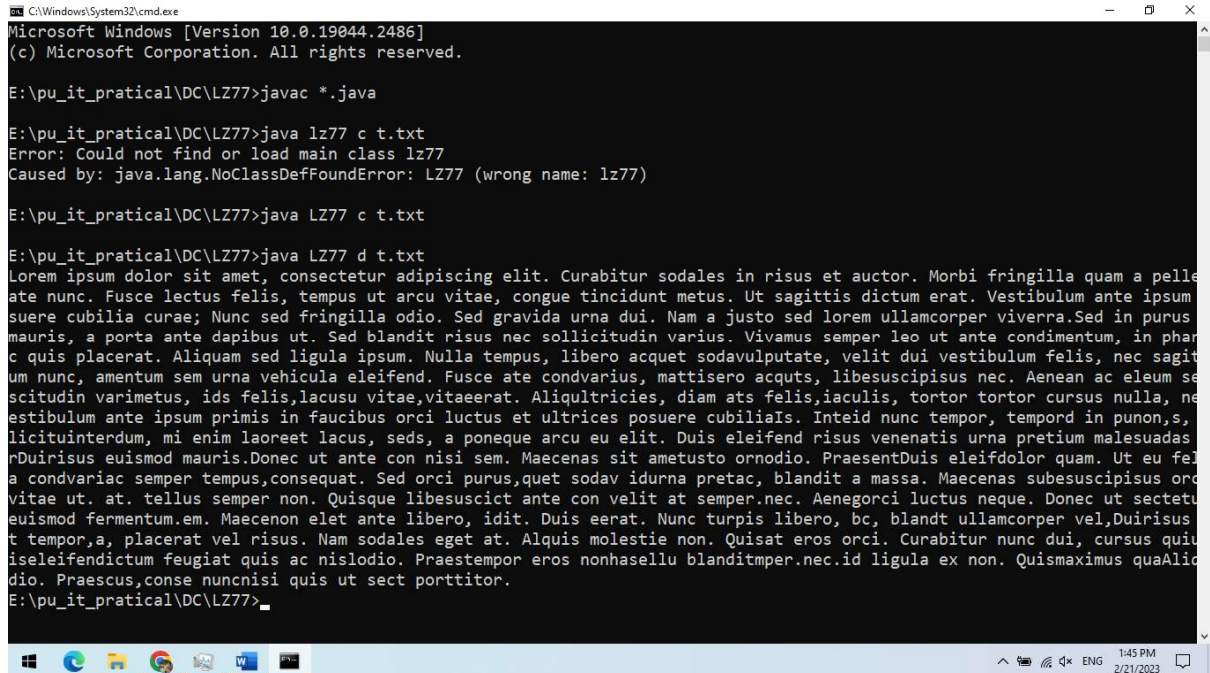
```

```

        decompressed.append(dictionary.get(currentTag));    }
    dictionary.add(nextDic.toString());
    nextDic = new StringBuilder();
    nextDic.append(dictionary.get(currentTag));    }    }
    return decompressed.toString();    } }

```

### Output:



```

Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

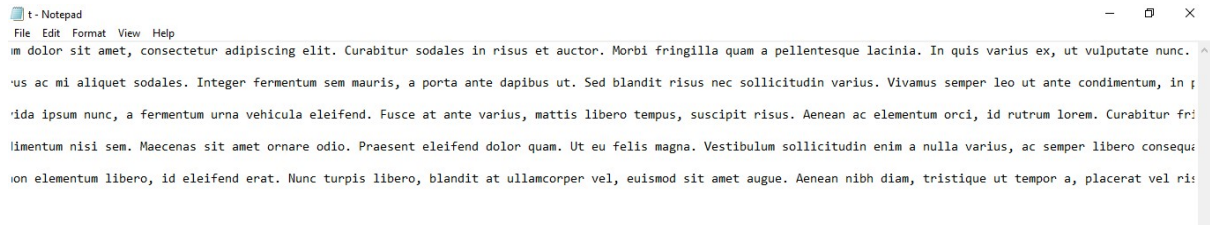
E:\pu_it_practical\DC\LZ77>javac *.java

E:\pu_it_practical\DC\LZ77>java lz77 c t.txt
Error: Could not find or load main class lz77
Caused by: java.lang.NoClassDefFoundError: LZ77 (wrong name: lz77)

E:\pu_it_practical\DC\LZ77>java LZ77 c t.txt

E:\pu_it_practical\DC\LZ77>java LZ77 d t.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur sodales in risus et auctor. Morbi fringilla quam a pelle
ate nunc. Fusce lectus felis, tempus ut arcu vitae, congue tincidunt metus. Ut sagittis dictum erat. Vestibulum ante ipsum
suere cubilia curae; Nunc sed fringilla odio. Sed gravida urna dui. Nam a justo sed lorem ullamcorper viverra. Sed in purus
mauris, a porta ante dapibus ut. Sed blandit risus nec sollicitudin varius. Vivamus semper leo ut ante condimentum, in phar
c quis placerat. Aliquam sed ligula ipsum. Nulla tempus, libero acquet sodavulputate, velit dui vestibulum felis, nec sagit
um nunc, amentum sem urna vehicula eleifend. Fusce ate condvarius, mattisero acquts, libesusicipis nec. Aenean ac eleum se
scitudin varimetis, ids felis, lacusu vitae, vitaerarat. Aliquultricies, diam ats felis, iaculis, tortor tortor cursus nulla, ne
estibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubiliaIs. Inteid nunc tempor, tempord in punon,s,
licituinterdum, mi enim laoreet lacus, seds, a poneque arcu eu elit. Duis eleifend risus venenatis urna pretium malesuadas
rDuiris euismod mauris. Donec ut ante con nisi sem. Maecenas sit ametusto ornodio. PraesentDuis eleifdolor quam. Ut eu fel
a condvariac semper tempus, consequat. Sed orci purus, quet sodav idurna pretac, blandit a massa. Maecenas subesuscipis ord
vitae ut. at. tellus semper non. Quisque libesusiciat ante con velit at semper. nec. Aenagorci luctus neque. Donec ut sectetu
euismod fermentum. em. Maecenon elet ante libero, idit. Duis eerat. Nunc turpis libero, bc, blandt ullamcorper vel, Duiris
t tempor, a, placerat vel risus. Nam sodales eget at. Alquis molestie non. Quisat eros orci. Curabitur nunc dui, cursus qui
iseleifendictum feugiat quis ac nislodio. Praestempor eros nonhasellu blanditmpet. nec. id ligula ex non. Quismaximus quaAlid
dio. Praescus, conse nuncnisi quis ut sect porttitor.
E:\pu_it_practical\DC\LZ77>

```



```

File Edit Format View Help
im dolor sit amet, consectetur adipiscing elit. Curabitur sodales in risus et auctor. Morbi fringilla quam a pellentesque lacinia. In quis varius ex, ut vulputate nunc.
us ac mi aliquet sodales. Integer fermentum sem mauris, a porta ante dapibus ut. Sed blandit risus nec sollicitudin varius. Vivamus semper leo ut ante condimentum, in
ida ipsum nunc, a fermentum urna vehicula eleifend. Fusce at ante varius, mattis libero tempus, suscipit risus. Aenean ac elementum orci, id rutrum lorem. Curabitur fr
ilimentum nisi sem. Maecenas sit amet ornare odio. Praesent eleifend dolor quam. Ut eu felis magna. Vestibulum sollicitudin enim a nulla varius, ac semper libero consequ
ion elementum libero, id eleifend erat. Nunc turpis libero, blandit at ullamcorper vel, euismod sit amet augue. Aenean nibh diam, tristique ut tempor a, placerat vel ris

```

### t.txt file for compressing

Name	Date modified	Type	Size
LZ77.class	2/21/2023 1:43 PM	CLASS File	4 KB
LZ77	1/6/2023 3:37 PM	JAVA File	6 KB
ss	11/29/2022 4:29 PM	PNG File	135 KB
t	12/30/2022 4:23 PM	Text Document	3 KB
t.txt	2/21/2023 1:44 PM	LZ77 File	3 KB

### t.txt.LZ77 file generated while running

## PRACTICAL-7

**AIM: Write a program to Implement LZW algorithm.**

**Code:**

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
public class LZW {
    public static void main(String[] args) {
        try {
            if (args.length == 2) {
                File file = new File(args[1]);
                if (file.exists()) {
                    if (args[0].toLowerCase().contains("c")) { // Compress
                        String input = Files.readString(file.toPath());
                        byte[] compressedTags = compress(input);
                        for (byte tag : compressedTags)
                            {System.out.println("Tag<" + Byte.toUnsignedInt(tag) + ">");}
                        String newPath = file.getPath() + ".lzw";
                        File compressedFile = new File(newPath);
                        compressedFile.createNewFile();
                        Files.write(compressedFile.toPath(), compressedTags);
                    } else if (args[0].toLowerCase().contains("d")) { // Decompress
                        byte[] input = Files.readAllBytes(file.toPath());
                        String decompressedTxt = decompress(input);
                        System.out.println(decompressedTxt);
                        String newPath = file.getPath() + ".txt";
                        File decompressedFile = new File(newPath);
                        decompressedFile.createNewFile();
                        List<String> lines = new ArrayList<>();
                        lines.add(decompressedTxt);
                        Files.write(decompressedFile.toPath(), lines);
                    } else {
                        System.out.println(args[0] + " is invalid argument");
                    }
                } else {
                    System.out.println(args[1] + " is not an existing file");
                }
            } else {
                System.out.println("No arguments were supplied");
            }
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
    private static byte[] compress(String str) {
        ArrayList<String> dictionary = new ArrayList<>();
        List<Byte> tags = new ArrayList<>();
        for (int i = 0; i < str.length(); i++) {
            if (dictionary.size() == 128) {
```



```

        dictionary.clear();
    }
    int dictionaryIndex = -1;
    StringBuilder temp_str = new StringBuilder();
    temp_str.append(str.charAt(i));
    for (int j = i + 1; j < str.length(); j++){
        temp_str.append(str.charAt(j));
        if (dictionary.contains(temp_str.toString())) {
            dictionaryIndex = dictionary.indexOf(temp_str.toString());
            if (j == str.length() - 1) {
                byte tag = (byte) (dictionaryIndex + 128);
                tags.add(tag);
                i += temp_str.length();
            }
        }
        else {
            dictionary.add(temp_str.toString());
            byte tag = (byte) (dictionaryIndex + 128);
            if (dictionaryIndex == -1) {
                tag = (byte) str.charAt(i);
            }
            tags.add(tag);
            i += temp_str.length() - 1;
            break;
        }
    }
    if(str.length() - i == 1) {
        tags.add((byte) str.charAt(i));
        break;
    }
    byte[] bytes = new byte[tags.size()];
    for (int i = 0; i < tags.size(); i++) {
        bytes[i] = tags.get(i);
    }
    return bytes;
}

private static String decompress(byte[] input) {
    StringBuilder decompressed = new StringBuilder();
    ArrayList<String> dictionary = new ArrayList<>();
    StringBuilder nextDic = new StringBuilder();
    nextDic.append((char) input[0]);
    decompressed.append((char) input[0]);
    for (int i = 1; i < input.length; i++) {
        // Reset the dictionary
        if (dictionary.size() == 128) {
            dictionary.clear();
        }
        int currentTag = Byte.toUnsignedInt(input[i]) - 128;
        if(currentTag < 0) {
            nextDic.append((char) input[i]);
            decompressed.append((char) input[i]);
            dictionary.add(nextDic.toString());
            nextDic = new StringBuilder();
            nextDic.append((char) input[i]);
        }
        else {
            if(currentTag >= dictionary.size()) {
                nextDic.append(nextDic.substring(0, 1));
            }
        }
    }
}

```

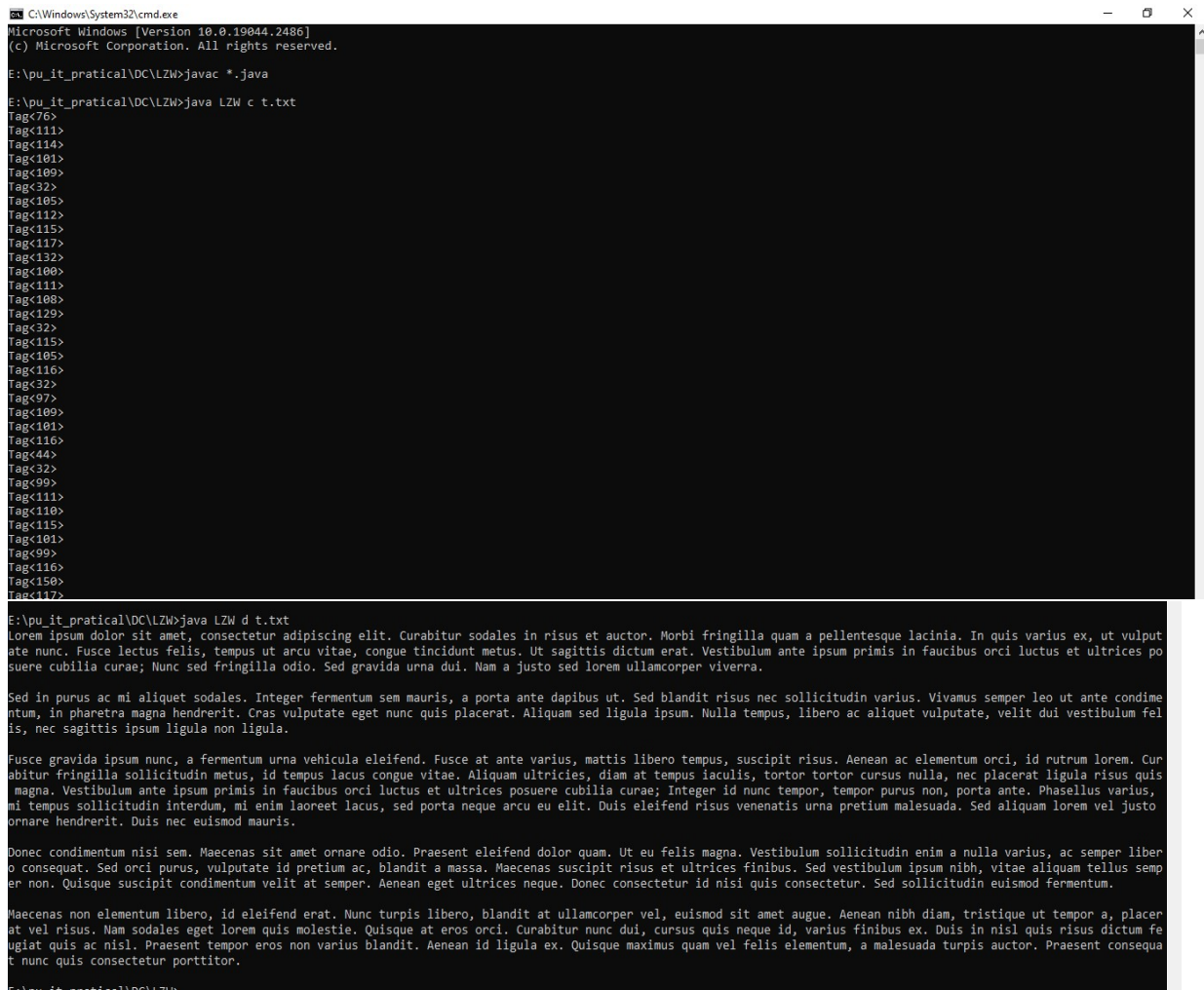


```

        decompressed.append(nextDic);
    }
    else {
        nextDic.append(dictionary.get(currentTag).charAt(0));
        decompressed.append(dictionary.get(currentTag));
    }
    dictionary.add(nextDic.toString());
    nextDic = new StringBuilder();
    nextDic.append(dictionary.get(currentTag));
}
return decompressed.toString();
}
}

```

### Output:



```

C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

E:\pu_it_practical\DC\LZW>javac *.java

E:\pu_it_practical\DC\LZW>java LZW c t.txt
Tag<76>
Tag<111>
Tag<114>
Tag<101>
Tag<109>
Tag<32>
Tag<105>
Tag<112>
Tag<115>
Tag<117>
Tag<132>
Tag<100>
Tag<111>
Tag<108>
Tag<129>
Tag<32>
Tag<115>
Tag<105>
Tag<116>
Tag<32>
Tag<97>
Tag<109>
Tag<101>
Tag<116>
Tag<44>
Tag<32>
Tag<99>
Tag<111>
Tag<110>
Tag<115>
Tag<101>
Tag<99>
Tag<116>
Tag<150>
Tag<117>

E:\pu_it_practical\DC\LZW>java LZW d t.txt
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur sodales in risus et auctor. Morbi fringilla quam a pellentesque lacinia. In quis varius ex, ut vulputate nunc. Fusce lectus felis, tempus ut arcu vitae, congue tincidunt metus. Ut sagittis dictum erat. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nunc sed fringilla odio. Sed gravida urna dui. Nam a justo sed lorem ullamcorper viverra.

Sed in purus ac mi aliquet sodales. Integer fermentum sem mauris, a porta ante dapibus ut. Sed blandit risus nec sollicitudin varius. Vivamus semper leo ut ante condimentum, in pharetra magna hendrerit. Cras vulputate eget nunc quis placerat. Aliquam sed ligula ipsum. Nulla tempus, libero ac aliquet vulputate, velit dui vestibulum felis, nec sagittis ipsum ligula non ligula.







Fusce gravida ipsum nunc, a fermentum urna vehicula eleifend. Fusce at ante varius, mattis libero tempus, suscipit risus. Aenean ac elementum orci, id rutrum lorem. Curabitur fringilla sollicitudin metus, id tempus lacus congue vitae. Aliquam ultricies, diam at tempus iaculis, tortor tortor cursus nulla, nec placerat ligula risus quis magna. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Integer id nunc tempor, tempor purus non, porta ante. Phasellus varius, mi tempus sollicitudin interdum, mi enim laoreet lacus, sed porta neque arcu eu elit. Duis eleifend risus venenatis urna pretium malesuada. Sed aliquam lorem vel justo ornare hendrerit. Duis nec euismod mauris.

Donec condimentum nisi sem. Maecenas sit amet ornare odio. Praesent eleifend dolor quam. Ut eu felis magna. Vestibulum sollicitudin enim a nulla varius, ac semper libero consequat. Sed orci purus, vulputate id pretium ac, blandit a massa. Maecenas suscipit risus et ultrices finibus. Sed vestibulum ipsum nibh, vitae aliquam tellus semper non. Quisque suscipit condimentum velit at semper. Aenean eget ultrices neque. Donec consectetur id nisi quis consectetur. Sed sollicitudin euismod fermentum.

Maecenas non elementum libero, id eleifend erat. Nunc turpis libero, blandit at ullamcorper vel, euismod sit amet augue. Aenean nibh diam, tristique ut tempor a, placerat vel risus. Nam sodales eget lorem quis molestie. Quisque at eros orci. Curabitur nunc dui, cursus quis neque id, varius finibus ex. Duis in nisl quis risus dictum feugiat quis ac nisl. Praesent tempor eros non varius blandit. Aenean id ligula ex. Quisque maximus quam vel felis elementum, a malesuada turpis auctor. Praesent consequat nunc quis consectetur porttitor.

E:\pu_it_practical\DC\LZW>

```

 LZW.class	2/21/2023 1:50 PM	CLASS File	4 KB
 LZW	1/9/2023 9:44 AM	JAVA File	6 KB
 ss	11/29/2022 4:29 PM	PNG File	135 KB
 t	12/30/2022 4:23 PM	Text Document	3 KB
 t.txt.lzw	2/21/2023 1:50 PM	LZW File	2 KB
 t.txt	2/21/2023 1:50 PM	Text Document	3 KB

## PRACTICAL-8

**AIM: Write a program to Implement LZ78 algorithm.**

**Code: LZW.java file**

```
import java.io.File;
import java.io.IOException;
import java.nio.file.Files;
import java.util.ArrayList;
import java.util.List;
public class LZ78 {
    public static void main(String[] args) {
        try {
            if (args.length == 2) {
                File file = new File(args[1]);
                if (file.exists()) {
                    if (args[0].toLowerCase().contains("c")) { // Compress
                        String input = Files.readString(file.toPath());
                        List<Tag> compressedTags = compress(input);
                        for (Tag tag : compressedTags) {
                            System.out.println("Tag<" + tag.getPosition() + ", " + tag.getNextChar() + ">");
                        }
                        byte[] compressedBytes = Tag.tagsToByteArray(compressedTags);
                        String newPath = file.getPath() + ".lz78";
                        File compressedFile = new File(newPath);
                        compressedFile.createNewFile();
                        Files.write(compressedFile.toPath(), compressedBytes);
                    } else if (args[0].toLowerCase().contains("d")) { // Decompress
                        byte[] input = Files.readAllBytes(file.toPath());
                        String decompressedTxt = decompress(Tag.bytesToTags(input));
                        System.out.println(decompressedTxt);
                        String newPath = file.getPath() + ".txt";
                        File decompressedFile = new File(newPath);
                        decompressedFile.createNewFile();
                        List<String> lines = new ArrayList<>();
                        lines.add(decompressedTxt);
                        Files.write(decompressedFile.toPath(), lines);
                    } else {
                        System.out.println(args[0] + " is invalid argument");
                    }
                } else {
                    System.out.println(args[1] + " is not an existing file");
                }
            } else {
                System.out.println("No arguments were supplied");
            }
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
    public static List<Tag> compress(String str){
        ArrayList<String> dictionary = new ArrayList<>();
        List<Tag> tags = new ArrayList<>();
        dictionary.add("\0");
        for (int i = 0; i < str.length(); i++) { // Reset the dictionary
            if (dictionary.size() == 256) {
```

```

        dictionary.clear();
        dictionary.add("\0");
    }
    int dictionaryIndex = 0;
    StringBuilder temp_str = new StringBuilder();
    for (int j = i; j < str.length(); j++){
        temp_str.append(str.charAt(j));
        if (dictionary.contains(temp_str.toString())) {
            dictionaryIndex = dictionary.indexOf(temp_str.toString());
            if (j == str.length() - 1) {
                Tag tag = new Tag((byte) dictionaryIndex, '\0');
                tags.add(tag);
                i += temp_str.length();
            }
        }
        else {
            char next = temp_str.charAt(temp_str.length() - 1);
            dictionary.add(temp_str.toString());
            Tag tag = new Tag((byte) dictionaryIndex, next);
            tags.add(tag);
            i += temp_str.length();
            break;
        }
    }
    return tags;
}

private static String decompress(List<Tag> tags) {
    ArrayList<String> dictionary = new ArrayList<>();
    StringBuilder decompressed = new StringBuilder();
    for (Tag tag : tags) {
        // Reset the dictionary
        if (dictionary.size() == 255) {
            dictionary.clear();
        }
        if (tag.getPosition() == 0) {
            dictionary.add(String.valueOf(tag.getNextChar()));
            if (tag.getNextChar() != 0)
                decompressed.append(tag.getNextChar());
        }
        else {
            StringBuilder tagBuilder = new StringBuilder();
            tagBuilder.append(dictionary.get(tag.getPosition() - 1));
            if (tag.getNextChar() != 0)
                tagBuilder.append(tag.getNextChar());
            dictionary.add(tagBuilder.toString());
            decompressed.append(tagBuilder);
        }
    }
    return decompressed.toString();
}

```

### Tag.java file

```

import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
public class Tag {
    private byte position;
    private char nextChar;

```



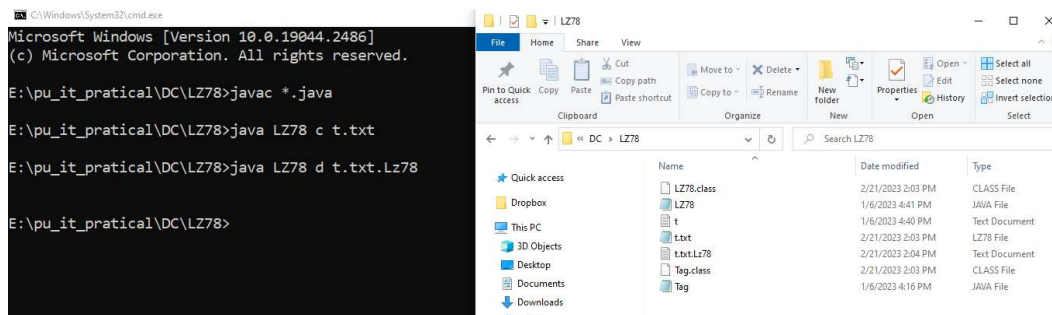
```
public Tag(byte position, char nextChar) {
    this.position = position;
    this.nextChar = nextChar; }
public Tag(byte[] tagBytes) {
    position = tagBytes[0];
    nextChar = (char) tagBytes[1]; }
public byte[] toByteArray() {
    byte[] bytes = new byte[2];
    bytes[0] = position;
    bytes[1] = (byte) nextChar;
    return bytes; }
public static byte[] tagsToByteArray(List<Tag> tags){
    byte[] result = new byte[tags.size() * 2];
    int tagsCounter = 0, resultCounter = 0;
    while (tagsCounter < tags.size()){
        byte[] tagByte = tags.get(tagsCounter).toByteArray();
        result[resultCounter] = tagByte[0];
        result[resultCounter + 1] = tagByte[1];
        resultCounter += 2;
        tagsCounter++; }
    return result; }
public static List<Tag> bytesToTags(byte[] bytes) {
    List<Tag> tags = new ArrayList<>(bytes.length / 2);
    for (int i = 0; i < bytes.length / 2; i++) {
        tags.add(new Tag(Arrays.copyOfRange(bytes, i*2, (i*2 + 2)))); }
    return tags; }
public int getPosition() { return Byte.toUnsignedInt(position); }
public void setPosition(int position) { this.position = (byte) position; }
public char getNextChar() { return nextChar; }
public void setNextChar(char nextChar) { this.nextChar = nextChar; }}
```

## Output:

### Before

LZ78	1/6/2023 4:41 PM	JAVA File	5 KB
t	1/6/2023 4:40 PM	Text Document	0 KB
Tag	1/6/2023 4:16 PM	JAVA File	2 KB

### After



The screenshot shows a Windows command prompt window with the following commands and output:

```
Microsoft Windows [Version 10.0.19044.2486]
(c) Microsoft Corporation. All rights reserved.

E:\pu_it_practical\DC\LZ78>javac *.java
E:\pu_it_practical\DC\LZ78>java LZ78 c t.txt
E:\pu_it_practical\DC\LZ78>java LZ78 d t.txt.LZ78
E:\pu_it_practical\DC\LZ78>
```

Next to the command prompt is a File Explorer window showing the contents of the 'LZ78' directory. The files listed are:

Name	Date modified	Type
LZ78.class	2/21/2023 2:03 PM	CLASS File
LZ78	1/6/2023 4:41 PM	JAVA File
t	1/6/2023 4:40 PM	Text Document
t.txt	2/21/2023 2:03 PM	LZ78 File
t.txt.LZ78	2/21/2023 2:04 PM	Text Document
Tag.class	2/21/2023 2:03 PM	CLASS File
Tag	1/6/2023 4:16 PM	JAVA File

## PRACTICAL-9

**AIM: Write a program which performs JPEG compression, process step by step for given 8x8 block and decompress.**

**Code:**

```
import numpy as np
quantization_matrix = np.array(
    [
        [16, 11, 10, 16, 24, 40, 51, 61],
        [12, 12, 14, 19, 26, 58, 60, 55],
        [14, 13, 16, 24, 40, 57, 69, 56],
        [14, 17, 22, 29, 51, 87, 80, 62],
        [18, 22, 37, 56, 68, 109, 103, 77],
        [24, 35, 55, 64, 81, 104, 113, 92],
        [49, 64, 78, 87, 103, 121, 120, 101],
        [72, 92, 95, 98, 112, 100, 103, 99],
    ]
)

def jpeg_compress(block):
    shifted_block = block - 128
    dct_block = shifted_block
    quantized_block = np.round(dct_block / quantization_matrix)
    quantized_block = quantized_block.flatten()
    return quantized_block

def jpeg_decompress(quantized_block):
    quantized_block = quantized_block.reshape((8, 8))
    dequantized_block = quantized_block * quantization_matrix
    idct_block = dequantized_block
    reconstructed_block = np.clip(idct_block + 128, 0, 255).astype(np.uint8)
    return reconstructed_block

def random_8x8():
    return np.random.randint(0, 256, size=(8, 8)).astype(np.uint8)

matrix = random_8x8()
print("Original Matrix:")
print(matrix)
compressed_matrix = jpeg_compress(matrix)
print("\nCompressed Matrix:")
print(compressed_matrix)
decompressed_matrix = jpeg_decompress(compressed_matrix)
print("\nDecompressed Matrix:")
print(decompressed_matrix)
```

## Output:

```
Original Matrix:
[[ 96 205 196 118 70 23 24 219]
 [231 252 232 246 232 84 136 154]
 [107 23 186 211 171 156 154 151]
 [250 160 227 184 175 140 173 101]
 [ 12 204 199 42 34 203 50 113]
 [ 36 36 70 203 68 31 133 154]
 [141 70 137 141 195 149 189 142]
 [ 64 175 43 35 23 83 220 219]]

Compressed Matrix:
[14. 7. 7. 15. 8. 4. 3. 1. 9. 10. 7. 6. 4. 4. 0. 0. 17. 12.
 4. 3. 1. 0. 0. 0. 9. 2. 4. 2. 1. 0. 1. 4. 8. 3. 2. 3.
 2. 1. 2. 3. 7. 5. 4. 1. 2. 2. 0. 0. 0. 3. 0. 0. 1. 0.
 1. 0. 3. 1. 2. 2. 1. 2. 1. 1.]

Decompressed Matrix:
[[255 205 198 255 255 255 255 189]
 [236 248 226 242 232 255 128 128]
 [255 255 192 200 168 128 128 128]
 [254 162 216 186 179 128 208 255]
 [255 194 202 255 255 237 255 255]
 [255 255 255 192 255 255 128 128]
 [128 255 128 128 231 128 248 128]
 [255 220 255 255 240 255 231 227]]

...Program finished with exit code 0
Press ENTER to exit console.
```

## PRACTICAL-10

**AIM: Write A Program To Implement Move-To-Front Algorithm.**

**Code:**

```
import java.util.*;
public class MoveToFront {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the text to encode: ");
        String text = scanner.nextLine();
        String encodedText = encode(text);
        System.out.println("Encoded text: " + encodedText);
        String decodedText = decode(encodedText);
        System.out.println("Decoded text: " + text);
    }
    public static String encode(String text) {
        List < Character > alphabet = new ArrayList < > ();
        for (int i = 0; i < 256; i++) {
            alphabet.add((char) i);
        }
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < text.length(); i++) {
            char c = text.charAt(i);
            int index = alphabet.indexOf(c);
            sb.append(index);
            alphabet.remove(index);
            alphabet.add(0, c);
        }
        return sb.toString();
    }
    public static String decode(String encodedText) {
        List < Character > alphabet = new ArrayList < > ();
        for (int i = 0; i < 256; i++) {
            alphabet.add((char) i);
        }
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < encodedText.length(); i++) {
            int index = Integer.parseInt(Character.toString(encodedText.charAt(i)));
            char c = alphabet.get(index);
            sb.append(c);
            alphabet.remove(index);
            alphabet.add(0, c);
        }
        return sb.toString();
    }
}
```

### Output:

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

E:\pu_it_practical\DC\mtf>javac *.java

E:\pu_it_practical\DC\mtf>java MoveToFront
Enter the text to encode: Hemil$Harishbhai$Chovatiya
Encoded text: 7210110910610941510111451151091051547774113118611771213
Decoded text: Hemil$Harishbhai$Chovatiya

E:\pu_it_practical\DC\mtf>_
```



## PRACTICAL-1

**AIM: Write A Program To Implement Burrows Wheeler Transform**

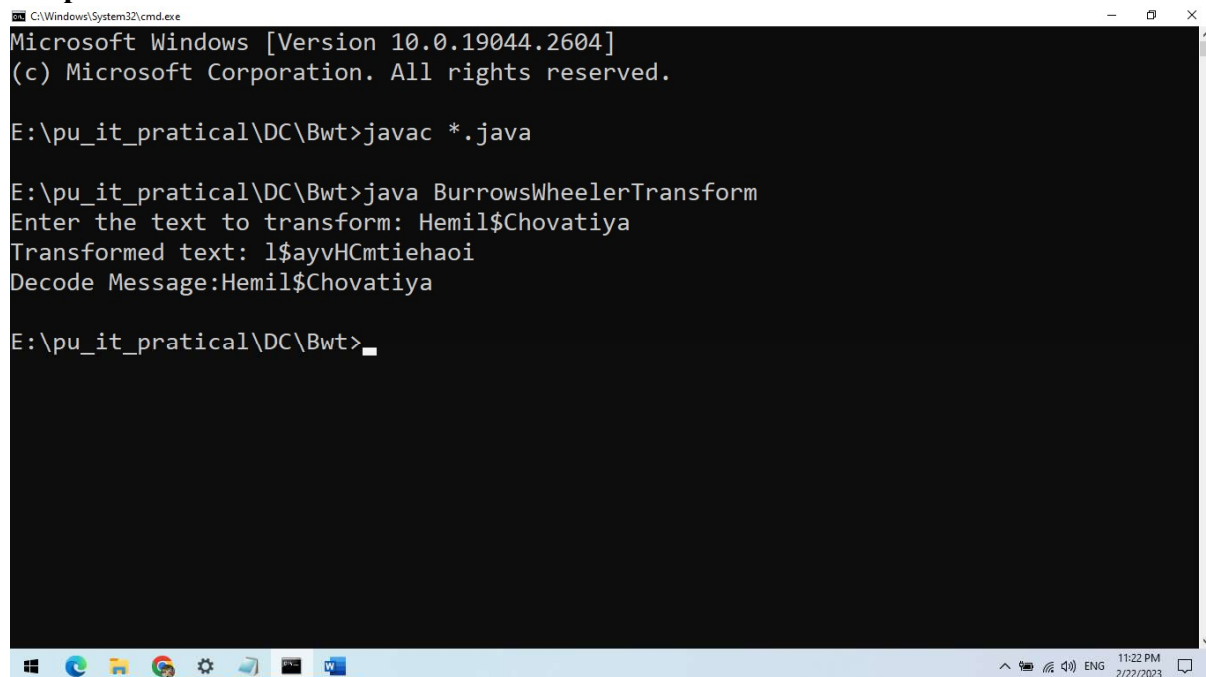
**Algorithm.**

**Code:**

```
import java.util.*;

public class BurrowsWheelerTransform {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the text to transform: ");
        String text = scanner.nextLine();
        String transformedText = transform(text);
        System.out.println("Transformed text: " + transformedText);
        System.out.println("Decode Message:" + text);
    }
    public static String transform(String text) {
        int length = text.length();
        String[] rotations = new String[length];
        for (int i = 0; i < length; i++) { rotations[i] = text.substring(i) + text.substring(0, i); }
        Arrays.sort(rotations);
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < length; i++) { sb.append(rotations[i].charAt(length - 1)); }
        return sb.toString();
    }
}
```

**Output:**



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2604]
(c) Microsoft Corporation. All rights reserved.

E:\pu_it_practical\DC\Bwt>javac *.java

E:\pu_it_practical\DC\Bwt>java BurrowsWheelerTransform
Enter the text to transform: Hemil$Chovatiya
Transformed text: l$ayvHCmtiehaoi
Decode Message:Hemil$Chovatiya

E:\pu_it_practical\DC\Bwt>
```