Date:

PRACTICAL-1

Aim: Write a program to perform pattern matching.

Code:

```
import java.util.Scanner;
public class DCDR2 {
 public static void main(String[] args) {
char x='n';
int pos= 0;
int k=0;
   Scanner scan = new Scanner (System.in);
   System.out.println("Enter Your String");
   String str=scan.nextLine();
   System.out.println("Enter Your pattern String");
   Scanner scan1 = new Scanner (System.in);
   String ptr=scan1.nextLine();
   for(int i=0;i<=str.length()-ptr.length();i++)
    {
    for(int j=0;j<ptr.length();j++)
    {
    if(str.charAt(i+j)!= ptr.charAt(j))
    break;
    if(j == ptr.length()-1)
    System.out.println("String Found at "+(i+1));
```

```
    k=i;
    }
    }
if(k>str.length()-ptr.length())
{
    System.out.print("no match found");
}
}
}
```

Output:

```
Enter Your String
my name is mayank
Enter Your pattern String
is
String Found at 9
```

Sign:

Date:

PRACTICAL-2

Aim: Write a program to generate all possible 4 letter words.

Code:

```
public class DCDR {
public static void main(String[] args) {
   for(char a='a';a<='z';a++) // for 1st
     {for(char b='a';b<='z';b++) //for 2nd
         {for(char c='a';c<='z';c++)
             {for(char d='a';d<='z';d++)
               {System.out.print(a);
                System.out.print(b);
                System.out.print(c);
                System.out.println(d);
               }
             }
         }}}}
```

Output:

AAAA

.

.

ZZZZ **Sign:**

Date:

PRACTICAL-3

Aim: Write a program to find he format and the size of the file.

Code:

```
import java.io.File;

public class DCDR3 {

   public static void main(String[] args) {
    try{
       String path="C:\\Users\\mayank\\Desktop\\mayank sem 5.pdf";
      File file1 = new File(path);
      System.out.println("size of file is = "+file1.length()+"bytes");
      int i= path.lastIndexOf('.');
      int p = path.lastIndexOf('\\');
      if(i>p)
      {
      String extension= path.substring(i+1);
      System.out.println("Extension of file is = "+extension);
      }
    }catch(Exception e)
    {System.out.println("File not Found");
    }
   }
}
```

Output:

```
size of file is = 56322 bytes
Extension of file is = pdf
```

Sign:

Date:

PRACTICAL-4

Aim: Write a program to calculate frequency and information content each character in string.

Code:

```
import java.util.Scanner;

public class DCDR4 {

   static final int SIZE = 26;
    public static void main(String[] args) {
        System.out.println("Enter your String (in small)");
        Scanner sc = new Scanner(System.in);
       String str =sc.next();

        printCharWithFreq(str);
    }

 static void printCharWithFreq(String str)
    {
        int n = str.length();
        int[] freq = new int[SIZE];
        for (int i = 0; i < n; i++)
            freq[str.charAt(i) - 'a']++;
        for (int i = 0; i < n; i++)
        {
            if (freq[str.charAt(i) - 'a'] != 0)
 { System.out.println("the Frequency of:" + str.charAt(i) + "is" +
freq[str.charAt(i) - 'a'] + " ");
                freq[str.charAt(i) - 'a'] = 0;
            }
        }
    }
```

```
}
```

Output:

```
Enter your String (in small)
jkjkdjd
the Frequency of:j is 3
the Frequency of:k is 2
the Frequency of:d is 2
```

Sign:

Date:

PRACTICAL-5

Aim: Study and implement Huffman code.

Code:

```
package dcdr;
import java.util.PriorityQueue;
import java.util.Scanner;
import java.util.Comparator;
class HuffmanNode {
	int data;
	char c;
	HuffmanNode left;
	HuffmanNode right;
}
class MyComparator implements Comparator<HuffmanNode> {
	public int compare(HuffmanNode x, HuffmanNode y)
	{
		return x.data - y.data;
	}
}
public class Practical5 {
	public static void printCode(HuffmanNode root, String s)
	{
		if (root.left
				== null
			&& root.right
				== null
			&& Character.isLetter(root.c)) {
			System.out.println(root.c + ":" + s);
			return;
		}
		printCode(root.left, s + "0");
		printCode(root.right, s + "1");
	}
	public static void main(String[] args)
	{
		Scanner s = new Scanner(System.in);
		int n = 6;
		char[] charArray = { 'a', 'b', 'c', 'd', 'e', 'f' };
```

```
        int[] charfreq = { 5, 9, 12, 13, 16, 45 };
            PriorityQueue<HuffmanNode> q
                = new PriorityQueue<HuffmanNode>(n, new MyComparator());
            for (int i = 0; i < n; i++) {
                HuffmanNode hn = new HuffmanNode();
                hn.c = charArray[i];
                hn.data = charfreq[i];
                hn.left = null;
                hn.right = null;
                q.add(hn);
            }
            HuffmanNode root = null;
            while (q.size() > 1) {
                HuffmanNode x = q.peek();
                q.poll();
                HuffmanNode y = q.peek();
                q.poll();
                HuffmanNode f = new HuffmanNode();
                f.data = x.data + y.data;
                f.c = '-';
                f.left = x;
                f.right = y;
                root = f;
                q.add(f);
            }
            printCode(root, "");
        }
    }
```

Output:

```
f:0
c:100
d:101
a:1100
b:1101
e:111
```

Sign:

Date:

PRACTICAL-6

Aim: Study and implement Run Length Coding.

Code:

```
import java.util.Scanner;
public class DCDR6 {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter string: ");
String str = sc.nextLine();
System.out.println("Input: " + str);
String compressed = "";
char ch=0;
int count=1;
for (int x = 0; x < str.length(); x++) {
if (ch == str.charAt(x)){
count = count + 1;
} else {
compressed = compressed + ch;
if(count != 1){
compressed = compressed + count;
}
ch = str.charAt(x);
count = 1;
}
}
compressed = compressed + ch;
if(count != 1){
compressed = compressed + count;
}
System.out.println("Compressed: " + compressed);
}
}
```

Output:

```
Enter string: aaabbcgg
Compressed:a3b2cg2
```

Sign:

Date:

PRACTICAL-7

Aim: Study and implement digram coding.

Code:

```
#include<iostream>
#include<string>
#include<vector>
usingnamespacestd;
typedefstructdiagram
{
stringentry;
stringcode;
}diagram;
intmain()
{
vector<diagram>my;
my.push_back({"a","000"});
my.push_back({"b","001"});
my.push_back({"c","010"});
my.push_back({"d","011"});
my.push_back({"r","100"});
my.push_back({"ab","101"});
my.push_back({"ac","110"});
my.push_back({"ad","111"});
cout<<"entry\tcode\n";
for(inti=0;i<my.size();i++)
cout<<my[i].entry<<"\t"<<my[i].code<<endl;
stringa="abacadrcbaabcdac";
cout<<a<<endl;
for(inti=0;i<a.length();i++)
{
intfound=-1;
for(intj=0;j<my.size();j++)
{
intlen=my[j].entry.length();
intcount=0;
for(intk=0;k<len;k++)
if(a[i+k]==my[j].entry[k])
count++;
if(count==len)
found=j;
```

```
}
if(found!=-1)
{
intlen=my[found].entry.length();
cout<<my[found].code;
}}}
```

Output:

```
Entry       code
A            000
B            001
C            010
D            100
R            011
H            101
AABCDRA
00000000101010001100 0
```

Sign:

Date:

PRACTICAL-8

Aim: Study and implement LZ77 algorithm.

Code:

```
import java.io.*;
import java.util.*;
public class DCDR8 {
public static void main(String args[]) throws FileNotFoundException,
IOException
{
String input=null;
try
{
FileInputStream file=new
FileInputStream("C:/Users/Mayank/Desktop/xyz.txt");
DataInputStream data=new DataInputStream(file);
BufferedReaderbr=new BufferedReader(new InputStreamReader(data));
input=br.readLine();
System.out.println(input);
FileOutputStream f1=new
FileOutputStream("C:/Users/Chirag/Desktop/abc.txt");
DataOutputStream d1=new DataOutputStream(f1);
int pointer=0,len1=0;
char o='0',nc='0';
String w,y=null,x=null;
int i,j=0,z,l=0;
for(i=0;i<=input.length();i++)
{
w=null;
w=input.substring(0,i);
j=i+1;
do{
y=input.substring(i,j);
z=w.lastIndexOf(y);
if(z!=-1)
{
```

```
l=z;
j++;
if(j>input.length())
{
i=input.length();
pointer=w.length()-l;
len1=y.length();// length of characters copied
nc='0';
d1.writeInt(pointer);
d1.writeInt(len1);
d1.writeChar(nc);
break;
}
}
else
{
i=w.length()+(y.length()-1);
if(y.length()==1)
{
pointer=0;
len1=0;
}
else
{
pointer=w.length()-l;
len1=y.length()-1;// length of characters copied
}
nc=y.charAt(y.length()-1);//next character
d1.writeInt(pointer);
d1.writeInt(len1);
d1.writeChar(nc);
break;
}
}while(z!=-1);
}
d1.close();
data.close();
}
catch (Exception ex)
{}
```

```
FileInputStream f2=new FileInputStream("C:/Users/Chirag/Desktop/abc.txt");
DataInputStream d2=new DataInputStream(f2);
BufferedReader b=new BufferedReader(new InputStreamReader(d2));
FileOutputStreamfos=new
FileOutputStream("C:/Users/Mayank/Desktop/abc1.txt");
DataOutputStream dos=new DataOutputStream(fos);
int p=0,len=0;
char n;
String decomp = "";
while(f2.available()!=0)
{
p=d2.readInt();//pointer
System.out.println(p);
len=d2.readInt();//length
System.out.println(len);
n=d2.readChar();//nextcharacter
System.out.println(n);
if(p==0)
decomp+=n;
else
{
if(n!='0')
decomp+=(decomp.substring((decomp.length()-p),((decomp.length()-
p)+len))+n);
else
decomp+=(decomp.substring((decomp.length()-p),((decomp.length()-p)+len)));
}}
dos.writeChars(decomp);
System.out.println(decomp);
b.close();
dos.close();
}}
```

Output:

```
aaaaaabbbbbccccc

0

0
```

a

1

1

a

3

3

b

1

1

b

2

2

c

1

1

c

2

2

0

aaaaaabbbbbccccc

Sign:

Date:

PRACTICAL-9

Aim: Study and implement LZ78 algorithm.

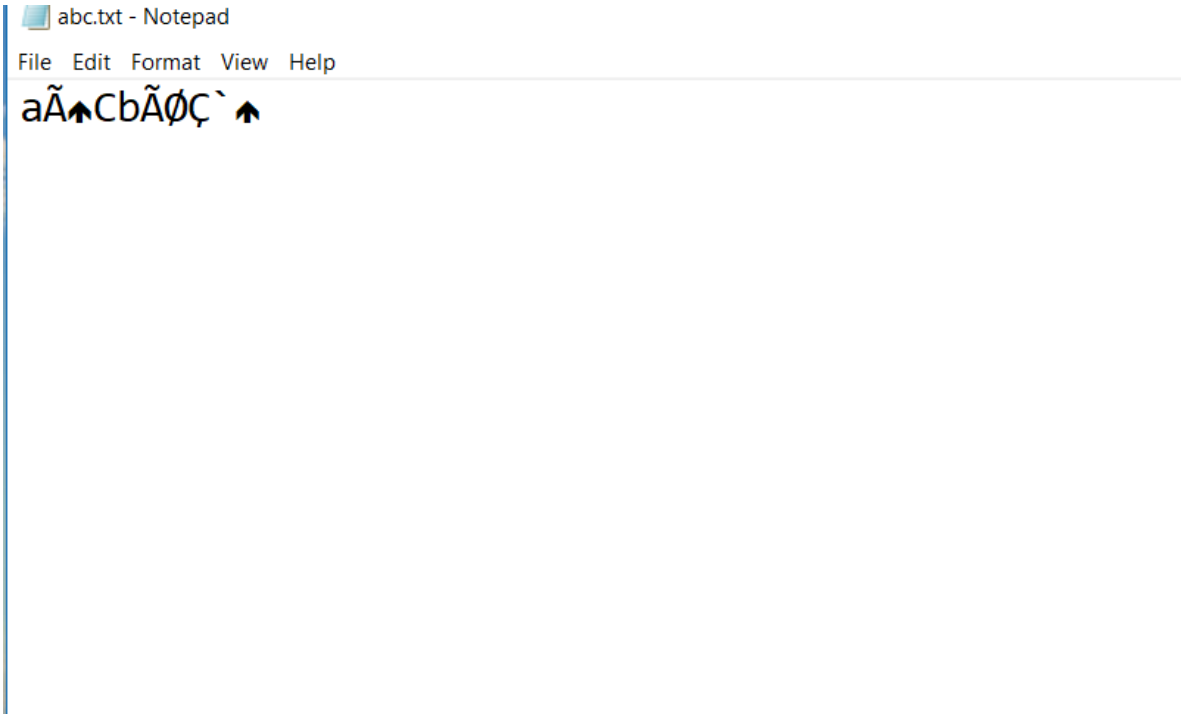
Code:

```
import java.io.*;
import java.util.*;
public class DCDR9 {
static long index = 0;
static byte addressLength = -1;
static long buffer;
static byte bufferLength = 0;
static Vector<Phrase>phraseStream = new Vector<Phrase>();
static BufferedReader reader;
static BufferedWriter writer;
static BufferedWriterlogWriter;
static String inPath = "C:/Users/Mayank/Desktop/xyz.txt";
static String outPath = "C:/Users/Mayank/Desktop/abc.txt";
static String logPath = "C:/Users/Mayank/Desktop/abc1.txt";
public static void main(String args[]) {
try {
reader = new BufferedReader(new FileReader(inPath));
writer = new BufferedWriter(new FileWriter(outPath));
logWriter = new BufferedWriter(new FileWriter(logPath));
encode();
reader.close(); writer.close(); logWriter.close();
} catch (IOException e) {
e.printStackTrace();}}
static class Phrase {
long code;
Phrase parent;
byte append;
HashMap children = new HashMap();
Phrase() {
init(null, 0);}
Phrase(Phrase phrase, int read) throws IOException {
init(phrase, read);
parent.children.put(new Byte(append),this);
write(); debug();}
```

```
void init(Phrase phrase, int read) {
phraseStream.add(this);
code = index;
if(((index-1) & (index-2)) == 0) addressLength++; index++;
parent = phrase;
append = (byte) read;}
void write() throws IOException {
buffer += parent.code<<bufferLength;
bufferLength+=addressLength;
while(bufferLength>= 8) {
writer.write((int)(buffer%256));
buffer>>=8;
bufferLength-=8;}
buffer += ((int)append)<<bufferLength;
writer.write((int)(buffer%256));
buffer>>=8;}
void debug() throws IOException {
logWriter.write(String.valueOf(code) + "\t" +
String.valueOf(addressLength) + "\t" +
String.valueOf(parent.code) + "\t" +
String.valueOf(append) + "\t");
Phrase curPhrase = this;
String str = "";
while(curPhrase != null) {
if(curPhrase.append == '\n') str = "~" + str;
else if(curPhrase.append == '\t') str = "=" + str;
else if(curPhrase.append == ' ') str = "_" + str;
else str = String.valueOf((char)curPhrase.append) + str;
curPhrase = curPhrase.parent;}
logWriter.write(str);
logWriter.newLine();}}
static void encode() throws IOException {
Phrase root = new Phrase();
int read = 0;
Phrase curPhrase = root;
while(true) {
read = reader.read();
if(read==-1) {
if(curPhrase != root)
new Phrase(curPhrase.parent, curPhrase.append);
```

```
writer.write((int)buffer);
return;}
if(curPhrase.children.containsKey(new Byte((byte)read)))
curPhrase = (Phrase) curPhrase.children.get(new Byte((byte)read));
else {
new Phrase(curPhrase, read);
curPhrase = root;
}}}}
```

Output:



Sign:

Date:

PRACTICAL-10

Aim: Study and implement LZW algorithm.

Code:

```
import java.io.*;
public class DCDR10 {
static String dict[];
static int point;
public DCDR10() {
}
public static String compress( String in ) {
dictInit( in.length() );
String out = "";
String w = "";
char k;
while (in.length() > 0) {
k = in.charAt(0);
in = in.substring(1);
if (dictHas(w + k)) { w+=k; }
else {
out+= dictCode(w);
dictAdd(w + k);
w = k + "";
}}
return out + dictCode(w);
}
public static String decompress( String in ) {
dictInit( in.length()+1 );
String out = "";
String w = "";
int k = (int)in.charAt(0);
w = dictChar( k );
in = in.substring(1);
out += w;
while (in.length() > 0) {
k = (int)in.charAt(0);
```

```
in = in.substring(1);
out += dictChar(k);
dictAdd(w + dictChar(k).charAt(0));
w = dictChar(k);
}
return out;
}
private static void dictInit( int size ) {
dict = new String[size];
point = 0;
}
private static booleandictHas( String s ) {
return (dictCode(s).length() > 0);
}
private static String dictCode( String s ) {
String code = "";
char c;
int i=0;
if (s.length() == 1) { //it's a character, just return the ascii code
return "" + s.charAt(0);
} else {
while ((code.length() == 0) && (i<dict.length)) {
if ((dict[i] != null) &&dict[i].equals(s)) {
c = (char)(i+256);
code = "" + c;
}
i++;
}
return code;
} }
private static String dictChar( int code ) {
if (code < 256) { return "" + (char)code; }
else if ((code-256) <dict.length) { return dict[code-256]; }
else { return ""; }
}
private static void dictAdd( String s ) {
dict[point] = s;
point++;
}
private static String getResponse() {
```

```
InputStreamReaderisr = new InputStreamReader( System.in );
BufferedReaderbr = new BufferedReader( isr );
String r = "n";
try {
r = br.readLine();
} catch ( IOExceptionioe ) {
// won't happen too often from the keyboard
}
return r;
}
public static void main( Stringargs[] ) {
System.out.print("Input: ");
String str1 = getResponse();
long tt = System.currentTimeMillis();
String c = compress( str1 );
tt = System.currentTimeMillis() - tt;
System.out.println("Compressed: (" + str1.length() + " -> " + c.length() + "
chars) in " + tt + " ms.");
if (args.length> 0) {
System.out.println("---COMPRESSED DATA FOLLOWS\r\n" + c + "\r\nEND
OF DATA---");
}
System.out.println("A saving of: " + (str1.length() - c.length()) + " chars.");
tt = System.currentTimeMillis();
String dec = decompress(c);
tt = System.currentTimeMillis() - tt;
System.out.println( "Decompressed (in " + tt + " ms.) to:\r\n" + dec );
}}
```

Output:

```
Input: aaaadfffggh
Compressed: (11 -> 9 chars) in 0 ms.
A saving of: 2 chars.
Decompressed (in 0 ms) to: aaaadfffggh
```

Sign:

Date:

PRACTICAL-11

Aim: Write a program to

a) Compress a text file using any compression technique.
b) Recompress a compressed file.
c) Find compression ratio in both steps and justify.

Code:

- **Compression:**

```
package dcdr2;
import java.io.*;
import java.util.zip.*;
public class DCDR11 {
public static void main(String args[]){
try{
FileInputStream fin=new
FileInputStream("C:/Users/Mayank/Documents/HTML files/prac.txt");
FileOutputStreamfout=new
FileOutputStream("C:/Users/Mayank/Documents/HTML files/com.txt");
DeflaterOutputStream out=new DeflaterOutputStream(fout);
int i;
while((i=fin.read())!=-1){
out.write((byte)i);
out.flush();
}
fin.close();
out.close();
}catch(Exception e){System.out.println(e);}
System.out.println("rest of the code");
}  }
```

- **Recompression:**

```
package dcdr2;
import java.io.*;
import java.util.zip.*;
public class DCDR11 {
public static void main(String args[]){
try{
FileInputStream fin=new FileInputStream("C:/Users/Chirag/Documents/HTML
files/com.txt");
FileOutputStreamfout=new
FileOutputStream("C:/Users/Chirag/Documents/HTML files/compression.txt");
```

```
DeflaterOutputStream out=new DeflaterOutputStream(fout);
int i;
while((i=fin.read())!=-1){
out.write((byte)i);
out.flush();
}
fin.close();
out.close();
}catch(Exception e){System.out.println(e);}
System.out.println("rest of the code");
}
}
```

Output:

Compression ratio of Compression: 3.8%

Compression ratio of Recompression: 0.98%

Sign:

Date:

PRACTICAL-12

Aim: Write a program to implement image compression and decompression using any one of above techniques.

Code:

```
import java.io.*;
import java.util.*;
import java.awt.image.*;
import javax.imageio.*;
import javax.imageio.stream.ImageOutputStream;
public class DCDR12 {
public static void main(String[] args) throws IOException {
File input = new File("C:/Users/Mayank/Pictures/Saved Pictures/hello.png");
BufferedImage image = ImageIO.read(input);
File compressedImageFile = new
File("C:/Users/Chirag/Desktop/compress.jpg");
OutputStreamos =new FileOutputStream(compressedImageFile);
Iterator<ImageWriter>writers =ImageIO.getImageWritersByFormatName("jpg");
ImageWriter writer = (ImageWriter) writers.next();
ImageOutputStreamios = ImageIO.createImageOutputStream(os);
writer.setOutput(ios);
ImageWriteParam param = writer.getDefaultWriteParam();
param.setCompressionMode(ImageWriteParam.MODE_EXPLICIT);
param.setCompressionQuality(0.05f);
writer.write(null, new IIOImage(image, null, null), param);
os.close();
ios.close();
writer.dispose();
}}
```

Output:

Compressed image:

SIGN:

Date:

PRACTICAL-13

Aim: Write a program to implement vector space model for xml retrieval.

Code:

```
importjava.io.File;
importjavax.xml.parsers.*;
importorg.xml.sax.*;
importorg.xml.sax.helpers.DefaultHandler;
public class XMLRet{
private String currentElement;
privateintbookCount = 1;
publicXMLRet() {
try {
SAXParserFactory factory = SAXParserFactory.newInstance();
SAXParsersaxParser = factory.newSAXParser();
saxParser.parse(new File("bookstore.xml"), new MyHandler());
    } catch (Exception e) {
e.printStackTrace();
    }}
public static void main(String args[]) {
newXMLRet();
  }
classMyHandler extends DefaultHandler {
    @Override
public void startElement(String uri, String localName, String qName,
        Attributes attributes) throws SAXException {
currentElement = qName;
if (currentElement.equals("book")) {
System.out.println("Book " + bookCount);
bookCount++;
        String isbn = attributes.getValue("ISBN");
System.out.println("\tISBN:\t" + isbn);
      }
    }
    @Override
public void endElement(String uri, String localName, String qName)
throwsSAXException {
currentElement = "";
```

```
    }
        @Override
public void characters(char[] chars, int start, int length) throws SAXException {
if (currentElement.equals("title")) {
System.out.println("\tTitle:\t" + new String(chars, start, length));
      } else if (currentElement.equals("author")) {
System.out.println("\tAuthor:\t" + new String(chars, start, length));
      }
    }
  }
}
```

Bookstore.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<bookstore>
<book ISBN="0123456001">
<title>Java For Dummies</title>
<author>Tan Ah Teck</author>
<category>Programming</category>
<year>2009</year>
<edition>7</edition>
<price>19.99</price>
</book>
<book ISBN="0123456010">
<title>The Complete Guide to Fishing</title>
<author>Bill Jones</author>
<author>James Cook</author>
<author>Mary Turing</author>
<category>Fishing</category>
<category>Leisure</category>
<language>French</language>
<year>2000</year>
<edition>2</edition>
<price>49.99</price>
</book>
</bookstore>
```

Output:

```
Book 1
        ISBN:0123456001
```

Title: Java For Dummies
Author: Tan Ah Teck

Book 2

ISBN:0123456010
Title: The Complete Guide to Fishing
Author: Bill Jones
Author: James Cook
Author: Mary Turing

Sign: