Assignment

Kirti Gupta
Kashyap Pratiksha

November 28, 2020

CODE package project;
import java.awt.Color;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Random;
import java.util.Scanner;

import javax.swing.JFrame;

import com.jogamp.opengl.GL2;
import com.jogamp.opengl.GLAutoDrawable;
import com.jogamp.opengl.GLCapabilities;
import com.jogamp.opengl.GLEventListener;
import com.jogamp.opengl.GLProfile;
import com.jogamp.opengl.awt.GLCanvas;

public class encoderJogl implements GLEventListener

static Scanner sn = new Scanner(System.in);
static int choice;
static int logic;
private static String result="";
private static String ans;
static ArrayList¡Integer¿ dataset=new ArrayList¡¿();

static void longestCommon(String result)
int max=0;
int temp=0;
for(int i=0;i¡result.length();i++)

```
if(result.charAt(i)=='0')

temp++;

else
if(temp¿max——max==0)
max=temp;


char[] fill = new char[max];
Arrays.fill(fill, '0');
ans = new String(fill);
System.out.println("Longest common substring of zeros is "+ans);

    static void printSubStr(String str, int low, int high)

System.out.println( str.substring( low, high + 1));


    static int longestPalSubstr(String str)


    int maxLength = 1;

    int start = 0;
int len = str.length();

    int low, high;

    for (int i = 1; i ¡ len; ++i)

    low = i - 1;
high = i;
while (low ¿= 0  high ¡ len  str.charAt(low) == str.charAt(high))
if (high - low + 1 ¿ maxLength)
start = low;
maxLength = high - low + 1;

–low;
++high;
```

```
    low = i - 1;
high = i + 1;
while (low ¿= 0  high ¡ len
str.charAt(low)
== str.charAt(high))
if (high - low + 1 ¿ maxLength)
start = low;
maxLength = high - low + 1;

–low;
++high;


System.out.print("Longest palindrome substring is: ");
printSubStr(str, start, start + maxLength - 1);

    return maxLength;

public static void main(String[] args)

    Random r=new Random();
System.out.println("Enter the required option to choose the process of input
taking");
System.out.println("1. User Input");
System.out.println("2. Random Binary Input");
int option=sn.nextInt();

    if(option==1)
System.out.println("Enter the Binary String");
result=sn.next();


    else

    System.out.println("Enter 1 if you want to fixed subsequence(size must
be greater then 8) otherwise press 2");
int k=sn.nextInt();
System.out.println("Enter the length of String");
int length=sn.nextInt();
int k1=0;
```

```
if(k==1)

System.out.println("Enter the length of fixed sequence(it may 4 or 8)");
k1=sn.nextInt();

    length-=k1;


    for(int i=0;i¡length;i++)
boolean res=r.nextBoolean();
if(res)
result+="1";
else
result+="0";
if(k==1i==length/2);

    if(k==1)
if(k1==8)
result=result.substring(0,length/2)+"00000000"+result.substring(length/2);
else
result=result.substring(0,length/2)+"0000"+result.substring(length/2);

    //LPS String str=String.valueOf(result);
System.out.println(result);
System.out.println("Length of palindrome String is " + longestPalSubstr(str));
longestCommon(result);
System.out.println("Select the type of Line Coding you want");
System.out.println("1. NRZ-L2. NRZ-I");
System.out.println("3. Manchester4. Differential Manchester");
System.out.println("5. Alternate Mark Invrsion (AMI)");
System.out.println("6. Modified AMI (Scrambling)");
choice=sn.nextInt();
encodeData(result,choice);

    final GLProfile profile = GLProfile.get(GLProfile.GL2);
GLCapabilities capabilities = new GLCapabilities(profile);

    // The canvas
final GLCanvas glcanvas = new GLCanvas(capabilities);
encoderJogl l = new encoderJogl();
glcanvas.addGLEventListener(l);
```

```
    glcanvas.setSize(1500, 300);
JFrame.setDefaultLookAndFeelDecorated(true);
//creating frame
final JFrame frame = new JFrame ("straight Line");
frame.getContentPane().setBackground(new Color(0.0f,0.0f,1.0f));
//frame.setOpacity(0.8f);
//adding canvas to frame

    frame.getContentPane().add(glcanvas);

    frame.setSize(frame.getContentPane().getPreferredSize());
frame.setVisible(true);

    public static void encodeData(String s,int choice)

    switch(choice)

    case 1:
for(int i=0;i¡result.length();i++)
if(result.charAt(i)=='0')
dataset.add(-1);

else
dataset.add(1);



    break;

    case 2:
int prev;
if(result.charAt(0)=='0')
dataset.add(-1);
prev=-1;

else
dataset.add(1);
prev=1;
```

```
    for(int i=1;i¡result.length();i++)
if(result.charAt(i)=='0')
dataset.add(prev);

else
if(prev==-1)
dataset.add(1);
prev=1;

else
dataset.add(-1);
prev=-1;



    break;

    case 3:
for(int i=0;i¡result.length();i++)
if(result.charAt(i)=='0')
dataset.add(-1);
dataset.add(1);

else
dataset.add(1);
dataset.add(-1);


    break;
case 4:
int prev1;
if(result.charAt(0)=='0')
dataset.add(-1);
dataset.add(1);
prev1=1;

else
dataset.add(1);
dataset.add(-1);
prev1=-1;
```

```
    for(int i=1;i¡result.length();i++)
if(result.charAt(i)=='1')

    dataset.add(prev1);
dataset.add(-prev1);
prev1=-prev1;

else
dataset.add(-prev1);
dataset.add(prev1);

    break;
case 5:
int prev2=1;
for(int i=0;i¡result.length();i++)
if(result.charAt(i)=='0')
dataset.add(0);

else
dataset.add(prev2);
prev2=-prev2;



    break;

    case 6:
dataset.clear();
System.out.println("Choose the type of Scrambling you want");
System.out.println("1. B8ZS (Bipolar with 8 zero subsitution)");
System.out.println("2. HDB3 (High Density Bipolar 3 zero)");
int ch=sn.nextInt();

    if(ch==1)
if(s.contains("00000000"))
s=s.replace("00000000", "000VB0VB");
int prev4=-1;
for(int i=0;i¡s.length();i++)
if(s.charAt(i)=='0')
dataset.add(0);
```

```
else if(s.charAt(i)=='1')
if(prev4==1)
dataset.add(-1);
prev4=-1;

else
dataset.add(1);
prev4=1;


else if(s.charAt(i)=='V')
if(prev4==1)
dataset.add(1);
prev4=1;

else
dataset.add(-1);
prev4=-1;


else if(s.charAt(i)=='B')
if(prev4==1)
dataset.add(-1);
prev4=-1;

else
dataset.add(1);
prev4=1;




    else if(ch==2)
int prev5=-1,count=0;
for(int i=0;i¡s.length();i++)
if(s.charAt(i)=='1')
count++;
if(prev5==1)
dataset.add(-1);
```

```
prev5=-1;

else
dataset.add(1);
prev5=1;


    else

    if(i+3¡s.length())
if(s.substring(i,i+4).equals("0000"))
if(counts=s.replaceFirst("0000", "B00V");

else
s=s.replaceFirst("0000", "000V");




    if(s.charAt(i)=='B')
if(prev5==1)
dataset.add(-1);
prev5=-1;

else
dataset.add(1);
prev5=1;

count++;

else if(s.charAt(i)=='V')
if(prev5==1)
dataset.add(1);
prev5=1;

else
dataset.add(-1);
prev5=-1;

count++;
```

else
dataset.add(0);

break;
default:
System.out.println("Wrong Choice");
break;

@Override

public void display(GLAutoDrawable arg0)
// TODO Auto-generated method stub
final GL2 gl = arg0.getGL().getGL2();

for(float i=-1.0f;i¡=1.0f;i=i+0.5f)
gl.glBegin (GL2.GL$_L$$INES$);
$gl.glColor3f(0.0f, 0.0f, 0.0f)$;
$gl.glVertex3f(-1.0f, i, 0)$;
$gl.glVertex3f(1.0f, i, 0)$;
$gl.glEnd()$;
$for(floati = -1.0f; i <= 1.0f; i = i+0.1f)gl.glBegin(GL2.GL_LINES); //staticfieldgl.glColor$

if (choice==3 —— choice==4)
float v=-1.0f;
for(int i=0;i¡dataset.size();i++)
gl.glHint(gl.GL$_L$$INE_LOOP$, $gl.GL_NICEST$);
$gl.glEnable(gl.GL_LINE_LOOP)$;
$gl.glLineWidth(4)$;
$gl.glBegin(GL2.GL_LINES)$;
$gl.glColor3f(0.0f, 1.0f, 0.0f)$;

gl.glVertex3f(dataset.get(i)/10+v,(float) ((float)dataset.get(i)/2.0),0);
gl.glVertex3f(dataset.get(i)/10+v+0.05f,(float) ((float)dataset.get(i)/2.0),0);
gl.glEnd();
v+=0.05f;

v=-1.0f;
for(int i=0;i¡dataset.size()-1;i++)

gl.glBegin (GL2.GL$_L INES$); //$staticfieldgl.glColor3f(0.0f, 1.0f, 0.0f)$;

gl.glVertex3f(dataset.get(i)/10+v+0.05f,(float) ((float)dataset.get(i)/2.0),0);
gl.glVertex3f(dataset.get(i)/10+v+0.05f,(float) ((float)dataset.get(i+1)/2.0),0);
gl.glEnd();
v+=0.05f;

else
float v=-1.0f;
for(int i=0;i¡dataset.size();i++)
gl.glHint(gl.GL$_L INE_L OOP, gl.GL_N ICEST$);
$gl.glEnable(gl.GL_L INE_L OOP)$;
$gl.glLineWidth(6)$;
$gl.glBegin(GL2.GL_L INES)$; //$staticfieldgl.glColor3f(0.0f, 1.0f, 0.0f)$;

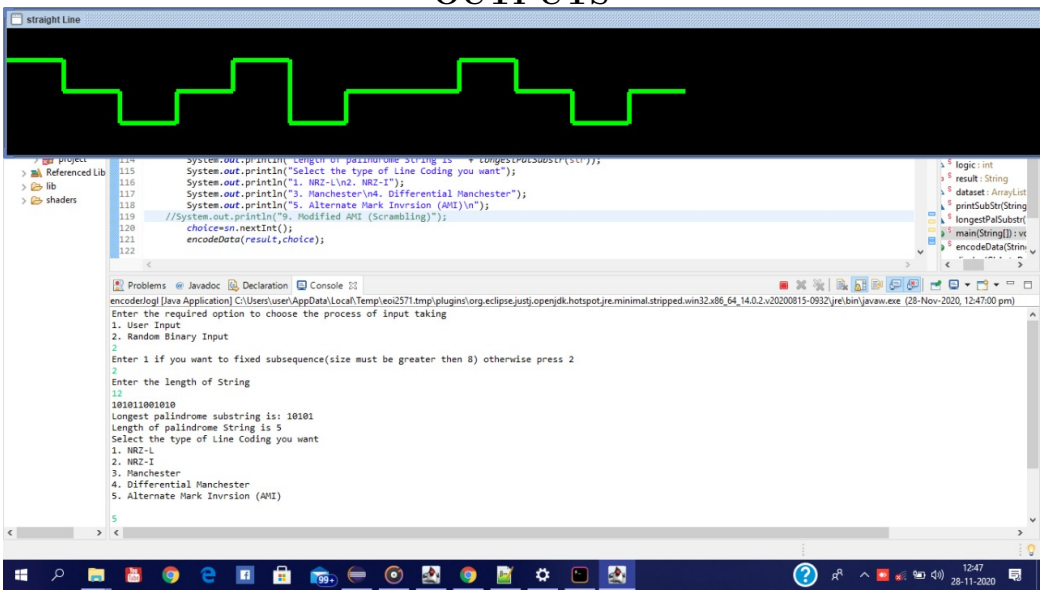
gl.glVertex3f(dataset.get(i)/10+v,(float) ((float)dataset.get(i)/2.0),0);
gl.glVertex3f(dataset.get(i)/10+v+0.1f,(float) ((float)dataset.get(i)/2.0),0);
gl.glEnd();
v+=0.1f;

v=-1.0f;
for(int i=0;i¡dataset.size()-1;i++)
gl.glBegin (GL2.GL$_L INES$); //$staticfieldgl.glColor3f(0.0f, 1.0f, 0.0f)$;

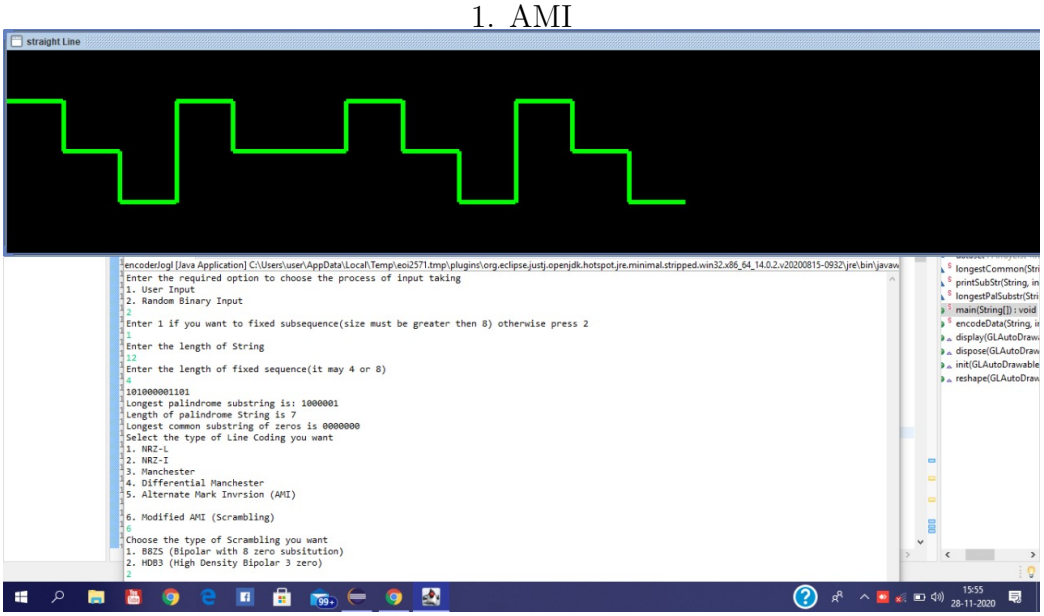
gl.glVertex3f(dataset.get(i)/10+v+0.1f,(float) ((float)dataset.get(i)/2.0),0);
gl.glVertex3f(dataset.get(i)/10+v+0.1f,(float) ((float)dataset.get(i+1)/2.0),0);
gl.glEnd();
v+=0.1f;

@Override public void dispose(GLAutoDrawable arg0) // TODO Auto-generated method stub
@Override public void init(GLAutoDrawable arg0) // TODO Auto-generated method stub
@Override public void reshape(GLAutoDrawable arg0, int arg1, int arg2, int arg3, int arg4) // TODO Auto-generated method stub

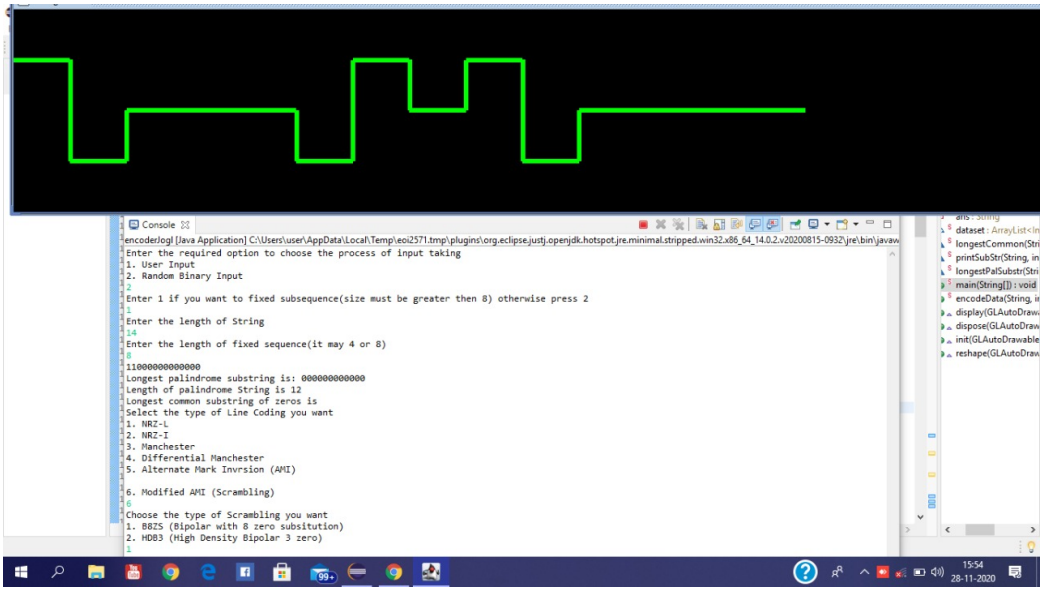
OUTPUTS



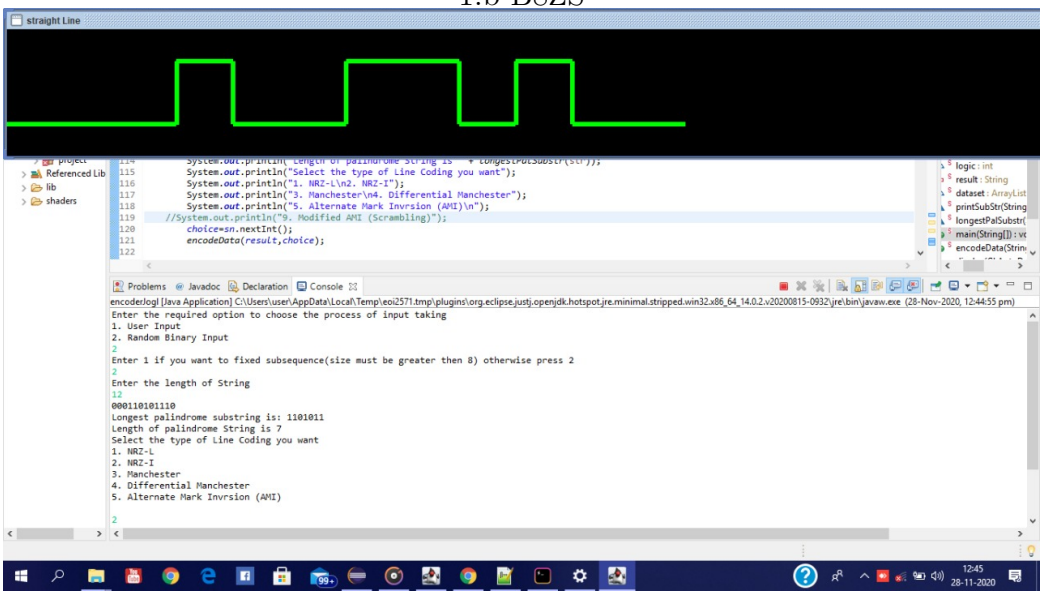
1. AMI



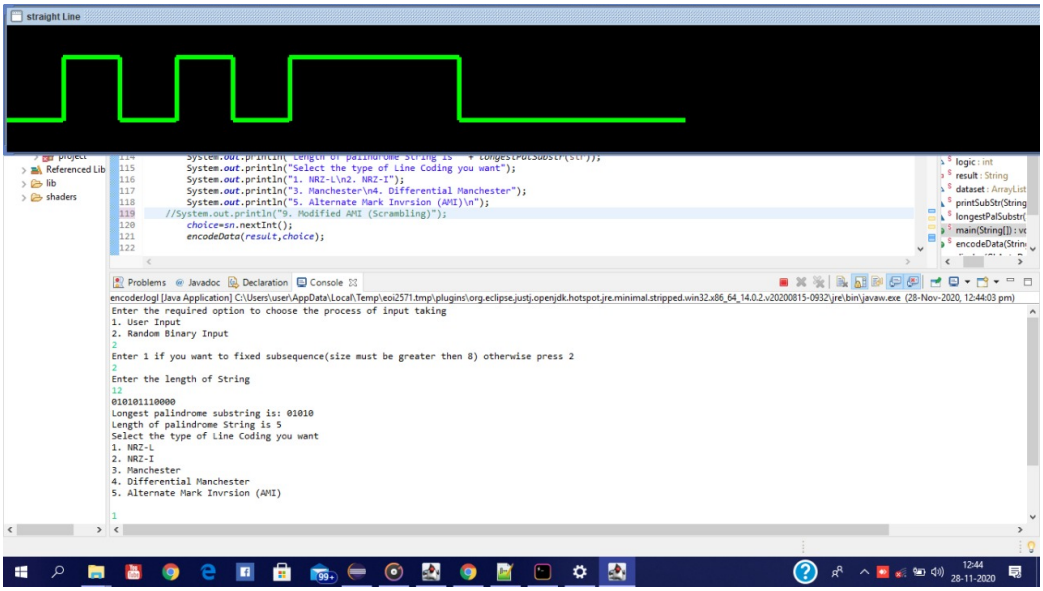
1.a HDB3



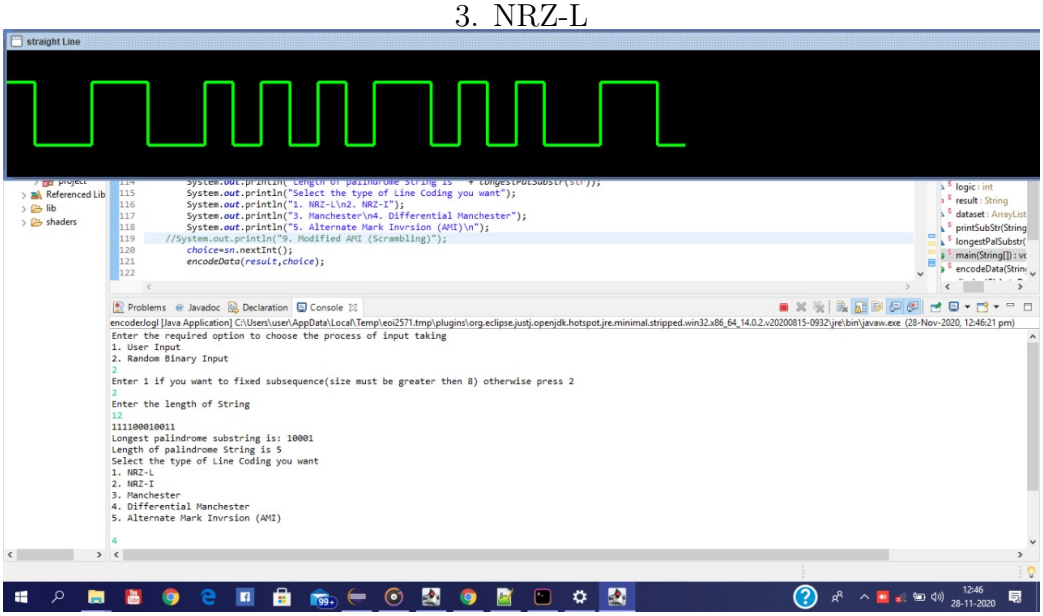
1.b B8ZS



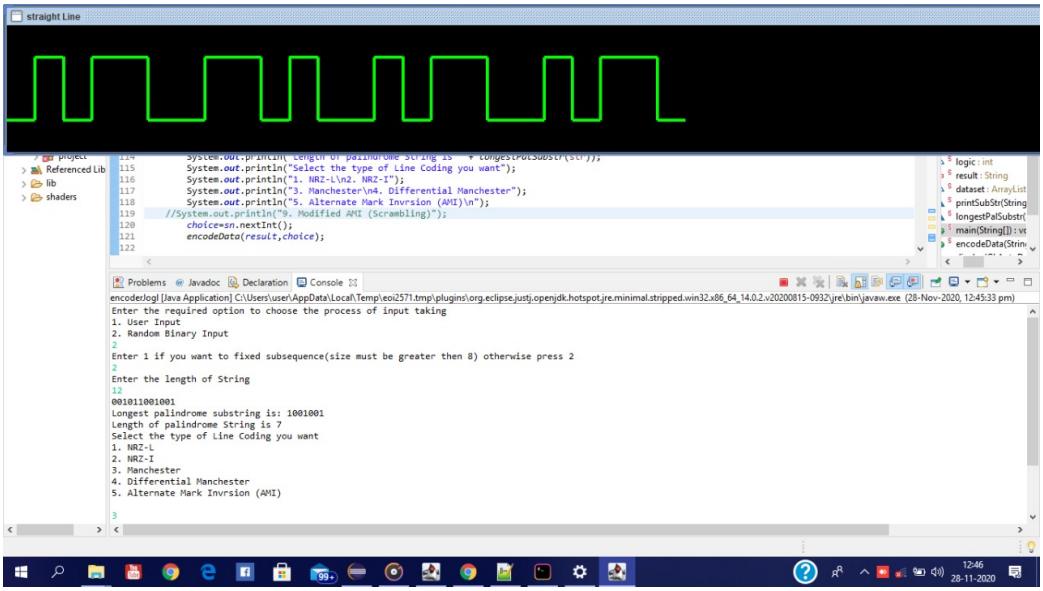
2. NRZ-I



3. NRZ-L



4. DIFFERENTIAL MANCHESTER



5. MANCHESTER

algorithm and references in LPS TIME COMPLEXITY : $O(n^2), where n is the length of input string.$
nested tranversal of the string is needed so time complexity is $o(n^2)$.
AUXILLIARY SPACE: O(1).no extra space is needed
REFERENCES : https://www.javatpoint.com/jogl-frame-using-swing
geeksforgeeks
https://www.tutorialspoint.com/jogl/drawing$_with_g_lines.htm$