Q.1 Border Layout: Implement following Example of Border Layout.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class BorderLayoutConverter extends JFrame {
   private JTextField inputField, resultField;
   private JButton binaryBtn, octalBtn, hexBtn;
   public BorderLayoutConverter() {
      setTitle("Number Converter");
      setSize(400, 200); setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new BorderLayout());
      JPanel northPanel = new JPanel();
      northPanel.add(new JLabel("Enter the number:"));
      inputField = new JTextField(10);
      northPanel.add(inputField);
      add(northPanel, BorderLayout.NORTH);
      JPanel centerPanel = new JPanel(new GridLayout(1, 3));
      binaryBtn = new JButton("Binary");
      octalBtn = new JButton("Octal");
      hexBtn = new JButton("Hex");
      centerPanel.add(binaryBtn);
      centerPanel.add(octalBtn);
      centerPanel.add(hexBtn);
      add(centerPanel, BorderLayout.CENTER);
      JPanel southPanel = new JPanel();
      southPanel.add(new JLabel("Result:"));
      resultField = new JTextField(15);
      resultField.setEditable(false);
      southPanel.add(resultField);
      add(southPanel, BorderLayout.SOUTH);
      binaryBtn.addActionListener(e -> convert("binary"));
      octalBtn.addActionListener(e -> convert("octal"));
      hexBtn.addActionListener(e -> convert("hex"));
      setVisible(true);
   }
   private void convert(String type) {
      try {
         int number = Integer.parseInt(inputField.getText().trim());
         switch (type) {
            case "binary": resultField.setText(Integer.toBinaryString(number));
               break;
            case "octal": resultField.setText(Integer.toOctalString(number));
               break;
            case "hex": resultField.setText(Integer.toHexString(number).toUpperCase());
               break;
         }
      } catch (NumberFormatException e) {
         resultField.setText("Invalid number");
      }
   }
   public static void main(String[] args) {
      new BorderLayoutConverter();
   }
}
```

Q.2 FlowLayout: Create a Java program using FlowLayout (aligned left, with horizontal gap 10px and vertical gap 20px) that adds three checkboxes labeled "Java", "Python", and "C++" into the frame.

```
import javax.swing.*;
import java.awt.*;
public class FlowLayoutExample extends JFrame {
   public FlowLayoutExample() {
      setTitle("FlowLayout Example");
      setSize(300, 150);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOS
E);
      setLayout(new FlowLayout(FlowLayout.LEFT,
10, 20));
      JCheckBox javaBox = new JCheckBox("Java");
      JCheckBox pythonBox = new
JCheckBox("Python");
      JCheckBox cppBox = new JCheckBox("C++");
      add(javaBox);
      add(pythonBox);
      add(cppBox);
      setVisible(true);
   }
   public static void main(String[] args) {
      new FlowLayoutExample();
   }
}
```

Q.3 GridLayout: Create a program that demonstrates the use of GridLayout. Display a 2x3 grid with the following numbers inside each box. Also, when the user clicks on any box, the number inside that box should swap with the number

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class GridSwapGame extends JFrame {
   private JButton[] buttons = new JButton[6];
   private JButton firstSelected = null;
   public GridSwapGame() {
      setTitle("GridLayout Swap Example");
      setSize(300, 200);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOS
E);
      setLayout(new GridLayout(2, 3));
      for (int i = 0; i < 6; i++) {
         buttons[i] = new JButton(String.valueOf(i +
1));
         add(buttons[i]);
         buttons[i].addActionListener(new
ActionListener() {
            public void actionPerformed(ActionEvent
e) {
               JButton clicked = (JButton)
e.getSource();
               handleSwap(clicked);
            }
         });
      }
      setVisible(true);
   }
   private void handleSwap(JButton clicked) {
      if (firstSelected == null) {
         firstSelected = clicked;
      } else {
         String temp = firstSelected.getText();
         firstSelected.setText(clicked.getText());
         clicked.setText(temp);
         firstSelected = null;
      }
   }
   public static void main(String[] args) {
      new GridSwapGame();
   }
}
```

Q. 4 Write a GUI program to find the factorial of a given number using applet. (You will need Java 8 to run applet)

```
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
public class FactorialApplet extends Applet
implements ActionListener {
   TextField inputField;
   Button calcButton;
   Label resultLabel;
   public void init() {
      setLayout(new FlowLayout());
      add(new Label("Enter a number:"));
      inputField = new TextField(10);
      add(inputField);
      calcButton = new Button("Calculate
Factorial");
      add(calcButton);
      calcButton.addActionListener(this);
      resultLabel = new Label("Result: ");
      add(resultLabel);
   }
   public void actionPerformed(ActionEvent e) {
      try {
         int num =
Integer.parseInt(inputField.getText());
         long fact = 1;
         for (int i = 1; i <= num; i++) {
            fact *= i;
         }
         resultLabel.setText("Result: " + fact);
      } catch (NumberFormatException ex) {
         resultLabel.setText("Invalid Input");
      }
   }
}
```