

Output:

Move disk 1 from slot A to slot C
Move disk 2 from slot A to slot B
Move disk 1 from slot C to slot B
Move disk 3 from slot B to slot C
Move disk 1 from slot C to slot A
Move disk 2 from slot B to slot C
Move disk 1 from slot A to slot C

Experiment - 9

Write a program for Tower of Hanoi :

```
class CrK {
```

```
static void TowerofHanoi(int n, char from_rod,  
char to_rod) throws exception
```

```
{
```

```
if(n==1){
```

```
System.out.println ("Move disk 1 from rod"  
+ "from rod + " + "to rod"  
+ " to rod");
```

```
return;
```

```
}
```

```
TowerofHanoi (n-1, from_rod, aux_rod, to_rod)
```

```
System.out.println ("Move disk " + n + " from rod"  
+ " from rod " + " to rod "  
+ " to rod );
```

```
TowerofHanoi (n-1, aux_rod, to_rod, from_rod);
```

```
}
```

```
public static void main (String args [] )
```

```
{ int n = 3;
```

```
TowerofHanoi (n, 'A', 'B', 'C');
```

```
}
```

```
}
```

Experiment-10

10.1) Red-coloured Box :

```

import javax.swing.*;
import java.awt.*;

public class RedBox extends JFrame {
    public RedBox() {
        setTitle("Red Box");
        setSize(200, 200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLocationRelativeTo(null);
        JPanel panel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                g.setColor(Color.Red);
                g.fillRect(50, 50, 100, 100);
            }
        };
        getContentPane().add(panel);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            RedBox redBox = new RedBox();
        });
    }
}

```

10.2) `import java.awt.*;`
`import java.awt.*;`

`public class BackgroundColorChanger extends JFrame {`

`private int red=0, green=0, blue=0;`

`public BackgroundColorChanger() {`

`setTitle("Background Color Changer");`

`setSize(400,400);`

`setDefaultCloseOperation(EXIT_ON_CLOSE);`

`colorPanel = new JPanel();`

`add(colorPanel, BorderLayout.CENTER);`

`changeBackgroundColor();`

`}`

`private void changeBackgroundColor() {`

`Timer timer = new Timer(1000, e → {`

`if (red < 255) {`

~~`red += 50;`~~

`}`

`else {`

`red = 0;`

`if (green < 255) {`

`green += 50;`

`}`

`else {`

`green = 0;`

`if (blue < 255) {`

```
blue += 50;
```

```
}
```

```
else {
```

```
blue = 0;
```

```
}
```

```
}
```

```
Dimension start();
```

```
}
```

```
public static void main (String [] args) {
```

```
new BackgroundColorChanger ().setVisible (true)
```

```
};
```

```
}
```