Exp No.9

Rishab Mandal

Batch: C23

Code:

Exp9DSSS.java :

```
import java.util.Arrays;

public class Exp9DSSS {

    public static void main(String[] args) {
        // Original data signal (binary representation)
        // System.out.print("Enter the length of data signal: ");
        int[] dataSignal = { 0, 1 };

        // Spreading code (PN sequence) Barker's code
        int[] spreadingCode = { 1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0 };

        // Spread the data signal using DSSS
        int[] spreadSignal = spreadDSSS(dataSignal, spreadingCode);

        // Display the results
```

```
        System.out.println("Original Data Signal: " +
Arrays.toString(dataSignal));
        System.out.println("Spreading Code (PN Sequence): " +
Arrays.toString(spreadingCode));
        System.out.println("Spread Signal: " + Arrays.toString(spreadSignal));

        // Recover the original signal by despread
        int[] recoveredSignal = despreadDSSS(spreadSignal, spreadingCode);

        // Display the recovered signal
        System.out.println("Recovered Signal: " +
Arrays.toString(recoveredSignal));
    }

    private static int[] spreadDSSS(int[] dataSignal, int[] spreadingCode) {
        int[] spreadSignal = new int[dataSignal.length *
spreadingCode.length];
        for (int i = 0; i < dataSignal.length; i++) {
            for (int j = 0; j < spreadingCode.length; j++) {
                spreadSignal[i * spreadingCode.length + j] = dataSignal[i] ^
spreadingCode[j];
            }
        }
        return spreadSignal;
    }
```

```
    private static int[] despreadDSSS(int[] spreadSignal, int[]
spreadingCode) {
        int length = spreadSignal.length / spreadingCode.length;
        int[] recoveredSignal = new int[length];


        for (int i = 0; i < length; i++) {
            int sum = 0;
            for (int j = 0; j < spreadingCode.length; j++) {
                sum += spreadSignal[i * spreadingCode.length + j] ^
spreadingCode[j];
            }
            System.out.print("Addition of " + " bit " + (i + 1) + " : " + sum);
            recoveredSignal[i] = (sum > 7) ? 1 : 0;
            if (sum > 7) {
                System.out.println(", Since sum is more than 7, it is converted to
1");
            } else
                System.out.println(", Since sum is less than 4, it is converted to
0");
        }


        return recoveredSignal;
    }
}
```

Output:

```
(base) PS C:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents> cd "c:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents\" ; if ($?) { javac Exp9DSSS.java } ; if ($?) { java Exp9DSSS }
Original Data Signal: [0, 1]
Spreading Code (PN Sequence): [1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0]
Spread Signal: [1, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1, 1]
Addition of  bit 1 : 0, Since sum is less than 4, it is converted to 0
Addition of  bit 2 : 11, Since sum is more than 7, it is converted to 1
Recovered Signal: [0, 1]
(base) PS C:\Users\Rishab\OneDrive\Desktop\MCC Exp Documents>
```