

MCC EXPERIMENT 8

```
import java.util.HashMap;
```

```
import java.util.Map;
```

```
public class CellularFrequencyReuseDemo {
```

```
    // Represents a single cell in the network
```

```
    static class Cell {
```

```
        int cellId;
```

```
        int frequencyChannel;
```

```
        public Cell(int cellId, int frequencyChannel) {
```

```
            this.cellId = cellId;
```

```
            this.frequencyChannel = frequencyChannel;
```

```
        }
```

```
    }
```

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

```
        // Create a network with multiple cells
```

```
        Cell[] cells = {
```

```
            new Cell(1, 1),
```

```
            new Cell(2, 2),
```

```
            new Cell(3, 1),
```

```
            new Cell(4, 2),
```

```
            new Cell(5, 3),
```

```
            new Cell(6, 1)
```

```
        };
```

```
        // Map to store frequency channels and their corresponding cells
```

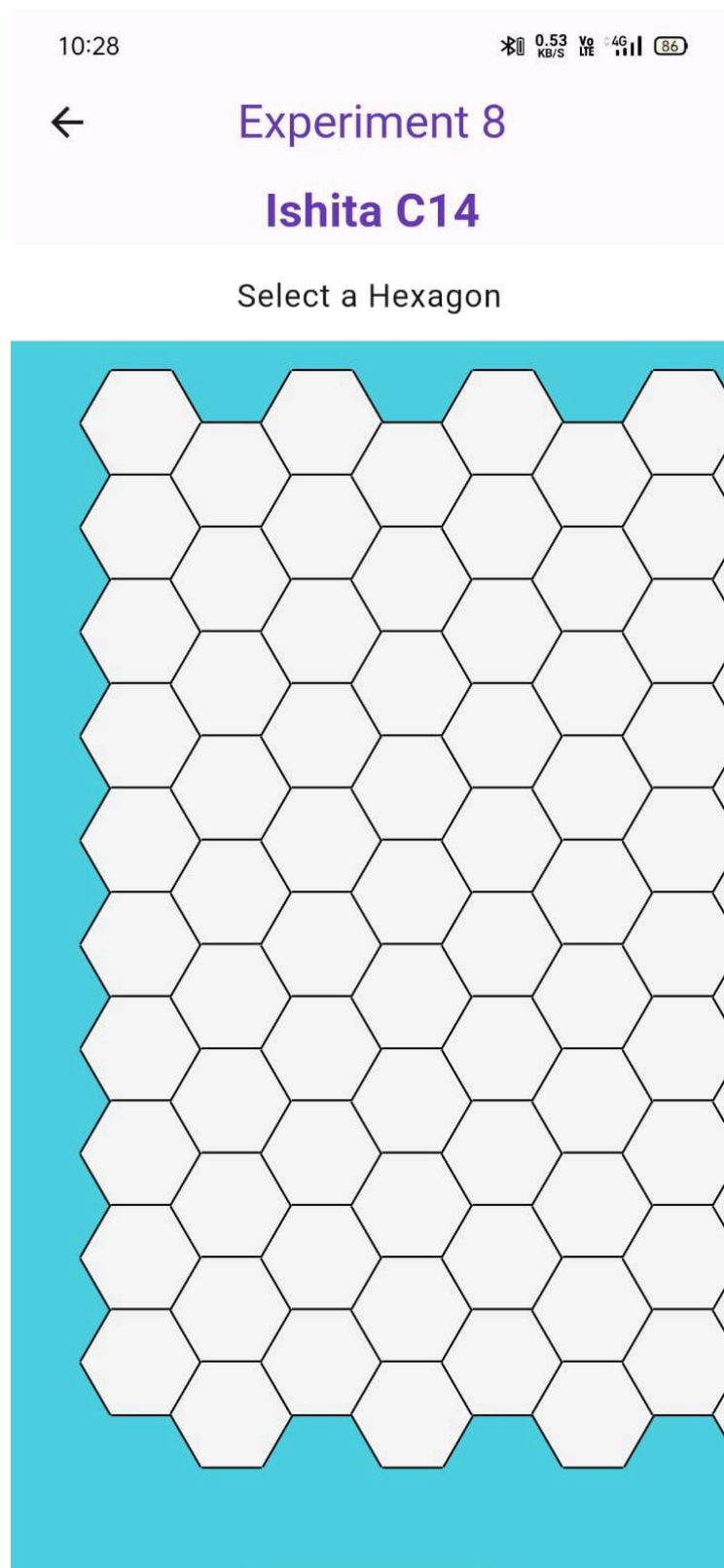
```
        Map<Integer, Cell> frequencyChannels = new HashMap<>();
```

```

// Demonstrate frequency reuse
for (Cell cell : cells) {
    // Check if the frequency channel is already allocated
    if (frequencyChannels.containsKey(cell.frequencyChannel)) {
        // If the channel is already allocated, print the reuse information
        System.out.println("Cell " + cell.cellId + " reuses frequency channel " +
cell.frequencyChannel +
        " (already allocated to Cell " +
frequencyChannels.get(cell.frequencyChannel).cellId + ")");
    } else {
        // If the channel is not allocated, allocate it to the current cell
        frequencyChannels.put(cell.frequencyChannel, cell);
        System.out.println("Cell " + cell.cellId + " uses frequency channel " + cell.frequencyChannel);
    }
}
}
}
}

```

Output:-



10:11 M

0.18 KB/S VoLTE 4G 89



Experiment 8

Ishita C14

Hexagon selected: 7,5

