```
12  public static void Run(bool useAnaconda = true)
13  {
14      //Get Training Data
15      var input = np.array(new float[,] { { 0, 0 }, { 0, 1 }, { 1, 0 }, { 1, 1 } });
16      var output = np.array(new float[] { 0, 1, 1, 0 });
17
18      //Create Model Structure
19      var model = new Sequential();
20      model.Add(new Dense(2));
21      model.Add(new Dense(32, activation: "relu"));
22      model.Add(new Dense(64, activation: "relu"));
23      model.Add(new Dense(1, activation: "sigmoid"));
24
25      model.Compile(optimizer: "sgd", loss: "binary_crossentropy", metrics: new[]
          { "accuracy" });
26
27      //Train model with the data
28      model.Fit(input, output, batch_size: 2, epochs: 1_000, verbose: 1);
29
30      //Make Prediction
31      Console.WriteLine($"[0, 0] = {Predict(model, 0, 0)}");
32      Console.WriteLine($"[0, 1] = {Predict(model, 0, 1)}");
33      Console.WriteLine($"[1, 0] = {Predict(model, 1, 0)}");
34      Console.WriteLine($"[1, 1] = {Predict(model, 1, 1)}");
35  }
36
37  public static int Predict(Sequential model, int p, int q)
38  {
39      var prediction = model.Predict(np.array(new float[,] { { p, q } }));
40      var predictedValue = prediction.GetData<float>().SingleOrDefault();
41      var roundedPredictedValue = (int)Math.Round(predictedValue);
42
43      return roundedPredictedValue;
44  }
```