

**PROGRAM:**

```
import math

def euclidean_distance(point1, point2):
    return math.sqrt((point1[0] - point2[0])**2 + (point1[1] - point2[1])**2)

def knn_classify(training_data, test_point, k=3):
    distances = [(euclidean_distance(test_point, data_point), label) for label, data_point
in training_data]
    sorted_distances = sorted(distances)[:k]
    label_counts = {0: 0, 1: 0}

    for distance, label in sorted_distances:
        label_counts[label] += 1

    return max(label_counts, key=label_counts.get)

def main():
    training_data = [(0, (1, 2)), (0, (2, 3)), (0, (3, 4)), (0, (4, 5)), (0, (5, 6)),
        (1, (6, 7)), (1, (7, 8)), (1, (8, 9)), (1, (9, 10)), (1, (10, 11))]
    test_point = (2, 4.5)
    k = 3
    result = knn_classify(training_data, test_point, k)
    if result == 0:
        print("The test point belongs to class 0.")
    else:
        print("The test point belongs to class 1.")

if __name__ == '__main__':
    main()
```

OUTPUT:

```
The test point belongs to class 1.
```