Test Cases	Output
def test_confidence_level_1(self): reading1=np.array([("knife",1, 0.99), ("scissor", 2, 0.65), ("spoon", 3, 0.33), ("spoon", 4, 0.80), ("keys", 5, 0.95)]) reading2=np.array([("keys", 5, 0.95), ("spoon", 4, 0.99), ("fork", 3, 0.99), ("scissor", 2, 0.95), ("knife",1, 0.55)]) expected =[('knife', 1, 0.99), ('scissor', 2, 0.95), ('fork', 3, 0.99), ('spoon', 4, 0.99), ('keys', 5, 0.95)] Result_obj=Robot() Result_obj.input(reading1,reading2) result =Result_obj.recognise() self.assertEqual(expected,result)	---------- Ran 1 test in 0.000s OK
def test_confidence_level_2(self): expected =None Result_obj=Robot() Result_obj.input() result =Result_obj.recognise() self.assertEqual(expected,result)	---------- Ran 1 test in 0.000s OK
def test_confidence_level_3(self): reading1=np.array([('knife',1, 0.99), ('scissor', 2, 0.65), ('spoon', 3, 0.33)]) reading2=None expected =[('knife',1, 0.99), ('scissor', 2, 0.65), ('spoon', 3, 0.33)] Result_obj=Robot() Result_obj.input(reading1) result =Result_obj.recognise() self.assertEqual(expected,result)	ValueError: shape mismatch: objects cannot be broadcast to a single shape ---------- Ran 1 test in 0.001s FAILED (errors=1) somesh@somesh-Aspire-E5-576G:~/Downloads/somesh_protyush_AST-dev/AST_Project$ python TestCase1.py [('knife', 1, 0.99), ('scissor', 2, 0.65), ('fork', 3, 0.99), ('spoon', 4, 0.99)] .[('knife', 1, 0.99)] [('knife', 1, 0.99), ('fork', 3, 0.99)]
def test_confidence_level_3(self): reading1=np.array([('knife',1, 0.99), ('scissor', 2, 0.65), ('spoon', 3, 0.33)]) reading2=np.array([('KNIFE',1, 0.99), ('SCISSOR', 2, 0.65), ('SPOON', 3, 0.33)]) expected =[('knife',1, 0.99), ('scissor', 2, 0.65), ('fork', 3, 0.33)] Result_obj=Robot() Result_obj.input(reading1,reading2) result =Result_obj.recognise() self.assertEqual(expected,result)	---------- Ran 1 test in 0.001s OK
def test_confidence_level_4(self): reading1=np.array([('knife',1, 0.99), ('scissor', 2, 0.65)]) reading2=np.array([('fork', 3, 0.99), ('spoon', 4, 0.99)]) expected =[('knife',1, 0.99), ('scissor', 2, 0.65),('fork', 3, 0.99), ('spoon', 4, 0.99)] Result_obj=Robot() Result_obj.input(reading1,reading2) result =Result_obj.recognise() self.assertEqual(expected,result)	---------- Ran 1 test in 0.000s OK
def test_confidence_level_5(self): reading1=np.array([('knife',1, 0.94),('knife',1, 0.69), ('knife',1, 0.89)]) reading2=np.array([('knife',1, 0.99), ('fork', 3, 0.99)]) expected=[('knife',1, 0.99),('fork', 3, 0.99)] Result_obj=Robot() Result_obj.input(reading1,reading2) result =Result_obj.recognise() self.assertEqual(expected,result)	---------- Ran 1 test in 0.001s OK

Ran 5 tests in 0.002s

OK