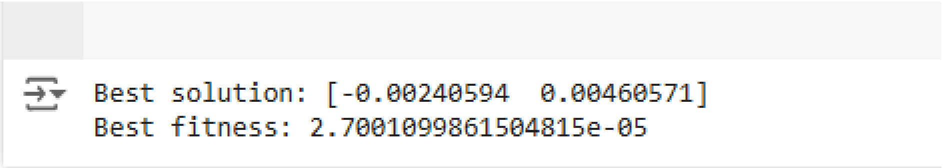
Lab 4

Cuckoo Search (CS) algorithm

code:

```
import numpy as np
def objective_function(x):
   return np.sum(x**2)
nests = 10  # Number of nests
dimension = 2  # Number of dimensions
iterations = 50  # Number of iterations
discovery_rate = 0.25  # Probability of worst nests being abandoned

population = np.random.uniform(-5, 5, (nests, dimension))
best_solution = population[0]
best_fitness = objective_function(best_solution)
for iteration in range(iterations):
    new_population = population + np.random.randn(nests, dimension)

     for i in range(nests):
      fitness = objective_function(new_population[i])


          if fitness < objective_function(population[i]):
         population[i] = new_population[i]


      if fitness < best_fitness:
            best_fitness = fitness

            best_solution = new_population[i]


        num_abandoned = int(discovery_rate * nests)

    worst_indices = np.argsort([objective_function(n) for n in population])[-num_abandoned:]

    population[worst_indices] = np.random.uniform(-5, 5, (num_abandoned, dimension))
```

output:



```
Best solution: [-0.00240594  0.00460571]
Best fitness: 2.7001099861504815e-05
```