BIS LAB 1

```
import random


def objective_function(x):
    return x ** 2


POP_SIZE = 20
GENS = 30
CROSSOVER_RATE = 0.8
MUTATION_RATE = 0.1
BOUNDS = [-10, 10]


def create_population(size):
    return [random.uniform(BOUNDS[0], BOUNDS[1]) for _ in range(size)]


def evaluate(population):
    return [objective_function(ind) for ind in population]


def select(population, fitness):
    i, j = random.sample(range(len(population)), 2)
    return population[i] if fitness[i] > fitness[j] else population[j]


def crossover(parent1, parent2):
    if random.random() < CROSSOVER_RATE:
        alpha = random.random()
        return alpha * parent1 + (1 - alpha) * parent2
    return parent1


def mutate(ind):
    if random.random() < MUTATION_RATE:
        ind += random.uniform(-1, 1)
```

```
        ind = max(min(ind, BOUNDS[1]), BOUNDS[0])
    return ind

def genetic_algorithm():
    population = create_population(POP_SIZE)
    for gen in range(GENS):
        fitness = evaluate(population)
        new_population = []
        for _ in range(POP_SIZE):
            parent1 = select(population, fitness)
            parent2 = select(population, fitness)
            child = crossover(parent1, parent2)
            child = mutate(child)
            new_population.append(child)
        population = new_population
        best_idx = fitness.index(max(fitness))
        best_solution = population[best_idx]
        best_fitness = fitness[best_idx]
        print(f"Gen {gen+1}: Best x = {best_solution:.4f}, f(x) = {best_fitness:.4f}")
    return best_solution, best_fitness

best_x, best_val = genetic_algorithm()
print("\nBest solution found:")
print(f"x = {best_x:.4f}, f(x) = {best_val:.4f}")
```

OUTPUT:

```
Gen 1: Best x = -6.6918, f(x) = 90.6965
Gen 2: Best x = 0.3788, f(x) = 90.6965
Gen 3: Best x = -6.9133, f(x) = 90.6965
Gen 4: Best x = -8.9913, f(x) = 90.6593
Gen 5: Best x = -9.0065, f(x) = 89.2660
Gen 6: Best x = -9.1510, f(x) = 89.2631
Gen 7: Best x = -9.3187, f(x) = 89.1403
Gen 8: Best x = -9.3628, f(x) = 89.1403
Gen 9: Best x = -9.4051, f(x) = 100.0000
Gen 10: Best x = -9.5598, f(x) = 100.0000
Gen 11: Best x = -9.7660, f(x) = 100.0000
Gen 12: Best x = -9.9976, f(x) = 100.0000
Gen 13: Best x = -9.9511, f(x) = 100.0000
Gen 14: Best x = -9.9928, f(x) = 100.0000
Gen 15: Best x = -10.0000, f(x) = 100.0000
Gen 16: Best x = -9.9981, f(x) = 100.0000
Gen 17: Best x = -9.9982, f(x) = 100.0000
Gen 18: Best x = -9.9992, f(x) = 100.0000
Gen 19: Best x = -9.9997, f(x) = 100.0000
Gen 20: Best x = -10.0000, f(x) = 100.0000
Gen 21: Best x = -9.9998, f(x) = 100.0000
Gen 22: Best x = -10.0000, f(x) = 100.0000
Gen 23: Best x = -10.0000, f(x) = 100.0000
Gen 24: Best x = -10.0000, f(x) = 100.0000
Gen 25: Best x = -9.4134, f(x) = 100.0000
Gen 26: Best x = -10.0000, f(x) = 100.0000
Gen 27: Best x = -9.2930, f(x) = 100.0000
Gen 28: Best x = -10.0000, f(x) = 100.0000
Gen 29: Best x = -10.0000, f(x) = 100.0000
Gen 30: Best x = -10.0000, f(x) = 100.0000

Best solution found:
x = -10.0000, f(x) = 100.0000
```