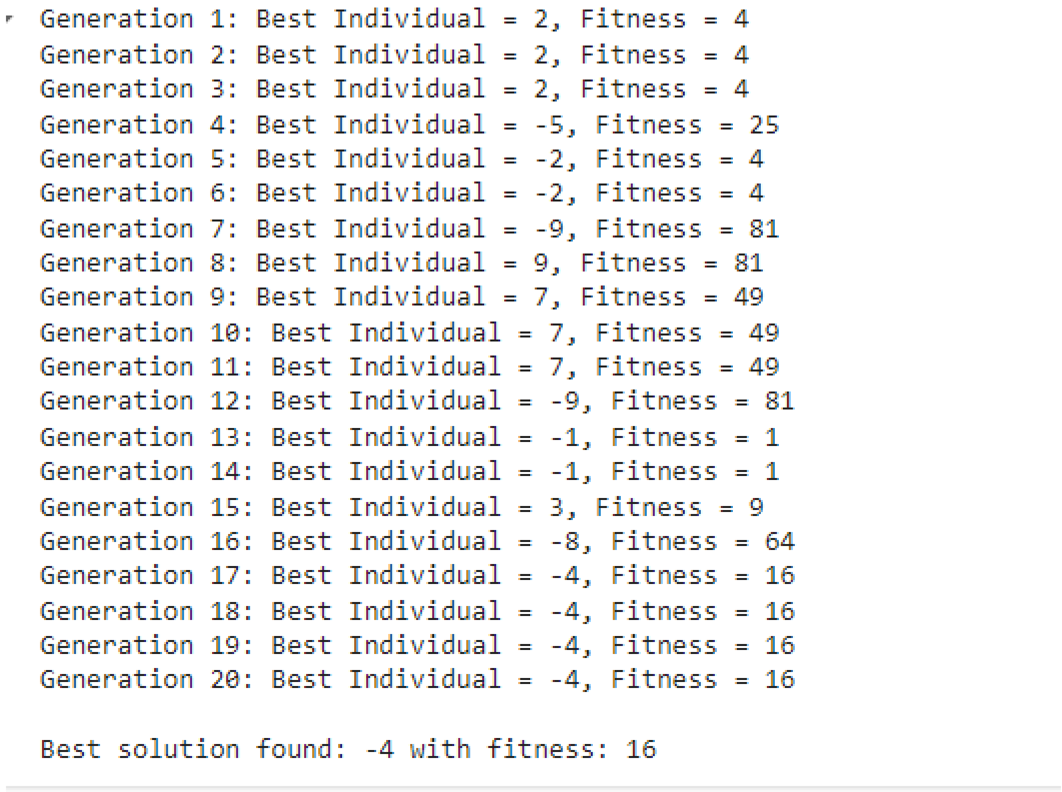
Lab 01:

Code : Genetic Algorithm Code

```python
import random

# Step 1: Define the problem - the function to maximize
def fitness_function(x):
    return x**2

# Step 2: Initialize parameters
population_size = 6
mutation_rate = 0.1
generations = 20
gene_pool = list(range(-10, 11))  # Possible values for genes

# Step 3: Create the initial population
def create_population(size):
    return [random.choice(gene_pool) for _ in range(size)]

# Step 4: Evaluate fitness of each individual
def evaluate_population(population):
    return [fitness_function(individual) for individual in population]

# Step 5: Selection - pick the two best individuals
def select_parents(population, fitnesses):
    sorted_population = sorted(zip(population, fitnesses), key=lambda x: x[1], reverse=True)
    return sorted_population[0][0], sorted_population[1][0]

# Step 6: Crossover - create a child from two parents
def crossover(parent1, parent2):
    return (parent1 + parent2) // 2  # Simple average crossover

# Step 7: Mutation - randomly change an individual
def mutate(individual):
    if random.random() < mutation_rate:
        return random.choice(gene_pool)
    return individual

# Step 8: Main loop for the Genetic Algorithm
population = create_population(population_size)

for generation in range(generations):
    fitnesses = evaluate_population(population)
    parent1, parent2 = select_parents(population, fitnesses)
    new_population = [crossover(parent1, parent2) for _ in range(population_size)]
    new_population = [mutate(individual) for individual in new_population]

    population = new_population
    best_individual = max(population, key=fitness_function)
    print(f"Generation {generation + 1}: Best Individual = {best_individual}, Fitness = {fitness_function(best_individual)}")

# Step 9: Output the best solution found
best_solution = max(population, key=fitness_function)
print(f"\nBest solution found: {best_solution} with fitness: {fitness_function(best_solution)}")
```

Output :



```
Generation 1: Best Individual = 2, Fitness = 4
Generation 2: Best Individual = 2, Fitness = 4
Generation 3: Best Individual = 2, Fitness = 4
Generation 4: Best Individual = -5, Fitness = 25
Generation 5: Best Individual = -2, Fitness = 4
Generation 6: Best Individual = -2, Fitness = 4
Generation 7: Best Individual = -9, Fitness = 81
Generation 8: Best Individual = 9, Fitness = 81
Generation 9: Best Individual = 7, Fitness = 49
Generation 10: Best Individual = 7, Fitness = 49
Generation 11: Best Individual = 7, Fitness = 49
Generation 12: Best Individual = -9, Fitness = 81
Generation 13: Best Individual = -1, Fitness = 1
Generation 14: Best Individual = -1, Fitness = 1
Generation 15: Best Individual = 3, Fitness = 9
Generation 16: Best Individual = -8, Fitness = 64
Generation 17: Best Individual = -4, Fitness = 16
Generation 18: Best Individual = -4, Fitness = 16
Generation 19: Best Individual = -4, Fitness = 16
Generation 20: Best Individual = -4, Fitness = 16

Best solution found: -4 with fitness: 16
```