## Theoretical Thinking

To implement a Genetic Algorithm (GA) for the chessboard placement problem described, you can follow these steps:

1. **Initialization**: Generate an initial population of chromosomes. Ensure that half of each type of chess piece is placed on the top half of the chessboard, and the other half on the bottom half as per the constraint.
2. **Fitness Evaluation**: Evaluate the fitness of each chromosome using the provided fitness function. This involves counting the number of conflicts and possibly applying penalties for uneven distribution of queens across columns.
3. **Selection**: Select individuals from the current population to form the mating pool for the next generation. Use selection methods like tournament selection, roulette wheel selection, or rank selection.
4. **Crossover**: Apply crossover to create offspring chromosomes from the selected individuals. Use a crossover probability of 0.8 to determine whether crossover should be applied or not.
5. **Mutation**: Apply mutation to the offspring chromosomes. Use a mutation probability of 0.1 to determine whether mutation should occur or not.
6. **Replacement**: Replace the current population with the new offspring population.
7. **Termination**: Repeat steps 2-6 for a certain number of generations defined by the user or until a termination condition is met (e.g., finding a solution with no conflicts).
8. **Solution Extraction**: Once the algorithm terminates, extract the best chromosome from the final population as the solution to the problem.

## How TP2. Genetic Algorithm for Chessboard Placement works?

1. **Initialization of Population**:
    - The algorithm begins by initializing a population of chromosomes, where each chromosome represents a possible arrangement of chess pieces on an 8x8 chessboard.
    - The size of the population is determined by the user input, and each chromosome is generated randomly.
2. **Fitness Evaluation**:
    - After initializing the population, the algorithm evaluates the fitness of each chromosome.

- Fitness is determined by counting the number of conflicts between pieces on the board. Lower conflict counts indicate higher fitness.
- The fitness function used is inversely proportional to the number of conflicts, meaning lower conflict counts result in higher fitness scores.

3. **Evolutionary Process**:
    - The algorithm iterates through a specified number of generations, aiming to improve the population over time.
    - During each generation, it applies selection, crossover, and mutation operators to produce the next generation of chromosomes.
    - **Selection**: Parents are probabilistically selected from the current population based on their fitness. Fitter chromosomes have a higher chance of being selected.
    - **Crossover**: Selected parents undergo crossover to create offspring. This process mixes genetic material from two parent chromosomes to generate new solutions.
    - **Mutation**: With a certain probability, mutation is applied to introduce random changes to the offspring chromosomes, ensuring diversity in the population.
4. **Selection Pressure**:
    - Selection pressure is maintained throughout the evolution process by favoring chromosomes with higher fitness during parent selection. This ensures that genetic material from better solutions is more likely to be inherited by the next generation.
5. **Termination**:
    - The algorithm repeats the evolutionary process for a specified number of generations, as determined by the user input.
6. **Identification of Best Solution**:
    - After the specified number of generations, the algorithm identifies the chromosome with the highest fitness in the final population. This chromosome represents the best solution found by the genetic algorithm.
7. **Printing the Best Solution**:
    - Finally, the algorithm prints the best solution found as a chessboard configuration, where each piece type is represented by its symbol.

In essence, the algorithm iteratively evolves a population of chessboard configurations using genetic operators. Through the evolutionary process, it seeks to improve the solutions by favoring those with fewer conflicts between pieces. The best solution found is determined based on the chromosome with the highest fitness in the final population.