Homework Assignment: Midterm Exam Scheduling with Capacity Constraint

Objective:

Implement a program to schedule midterm exams for given class lists and classroom capacities. Consider constraints such as student and professor availability, the ability to block specific hours for common courses, and the new constraint that if a classroom's capacity is N, only allow a midterm exam for a class whose capacity is at most N/2.

Input Files: There will be 2 input files.

1. Class List (CSV format):
    - Each row represents a Student ID, Professor Name, Course ID, and Exam Duration in minutes.

    Example:
    ```
    StudentID,Professor Name,CourseID,ExamDuration(in mins)
    1001,John Smith,CENG201,60
    1002,Jane Doe,MATH202,120
    ... (additional rows)
    ```

2. Classroom Capacities (CSV format):
    - Each row represents a classroom.
    - Columns: Room ID, Capacity.

    Example:
    ```
    RoomID,Capacity
    A301,30
    A302,25
    ... (additional rows)
    ```

Program Requirements:

Read the class list and classroom capacities from the provided CSV files.
Allow the user to input the hours that are blocked for other common courses.
Implement a scheduling algorithm to assign exam times for each class, adhering to the following constraints:

- A student cannot have two exams at the same time.
- A professor cannot have two exams at the same time.
- If a classroom's capacity is N, only allow a midterm exam for a class whose capacity is at most N/2.

- The exams are scheduled for 1 week between Monday through Saturday starting from 9:00AM to 6:00PM. If a schedule adhering the given constraints cannot be created with 6-day time frame, add one more day and reschedule exams.

Output the final schedule, including assigned exam times for each class, and any blocked hours.

Guidelines:

- Implement functions for reading CSV files, user input, and the scheduling algorithm.
- Use appropriate data structures to represent the class list, classroom capacities, and the schedule.
- Document your code with comments for clarity.
- Test your program with different inputs to ensure it handles various scenarios.

Example Output:

```
Exam Schedule:

Monday

9:00 AM - 10:30 AM: CENG101 - Room A301
11:00 AM - 12:30 PM: MATH102 - Room C302

... (additional schedule)

Tuesday

9:00 AM - 10:30 AM: EE213 - Room B303
9:00 AM - 11:00 AM: SENG211 - Room A301

... (additional schedule)

... (schedule for the remaining days)

Blocked Hours:

Wednesday

2:00 PM - 4:00 PM: Common Course (TİT102)
```

```
... (additional blocked hours)
```

Submission:

Submit a well-documented source code along with sample input files and a brief report explaining your implementation and any challenges you faced.

Hint: You may consider using Backtracking, Greedy Algorithms, Graph Coloring, and Simulated Annealing for this homework.