Practical No 9

**Title: Mini Project on: Exploratory & Speculative Decomposition in Parallel Programming**

Aim

1. Implement an exploratory decomposition mini-project (e.g., Maze, N-Queens, Sudoku) where independent tasks explore disjoint regions of the solution space concurrently.
2. Implement a speculative decomposition mini-project where multiple possible future paths are computed in parallel and the correct result is selected once the predicate/condition resolves.
3. Record and compare sequential vs. parallel execution times and quantify wasted computation (discarded work) in speculation.

Software/Hardware Requirements

Software: GCC/Clang with OpenMP (recommended) or OpenMPI/MPICH for MPI; Linux/Unix environment; plotting tool (e.g., gnuplot/Excel).  
Hardware: Multi-core CPU (recommended ≥4 cores). Optional: multi-node cluster for MPI.

Introduction

Parallel decomposition strategies divide work to exploit concurrency:

* Exploratory Decomposition: Partition a search/solution space into subspaces explored concurrently (e.g., tree branches in backtracking, frontier slices in graph search). Suited to irregular workloads like N-Queens, Sudoku, Maze traversal.
* Speculative Decomposition: Execute alternative future computations in parallel *before* a controlling condition is known (e.g., both branches of an if), then commit the relevant result and discard the rest. Highlights the trade-off between reduced latency and wasted work.

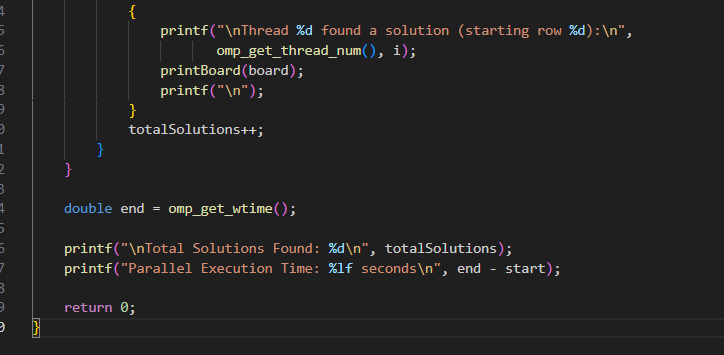
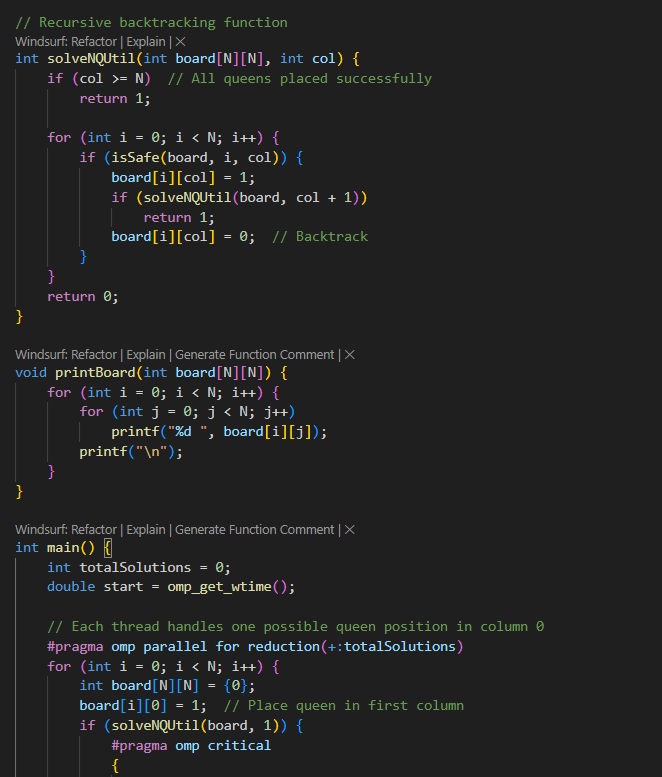
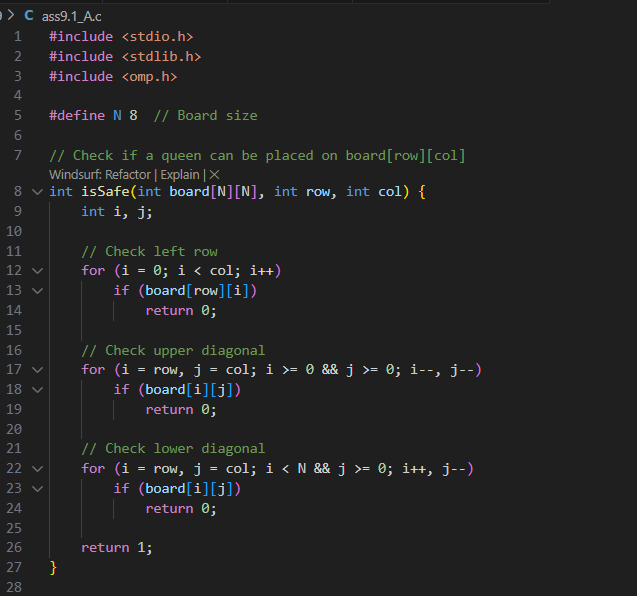
These techniques illuminate limits imposed by serial portions, synchronization, and overheads, reinforcing concepts like Amdahl’s Law and load balancing.

Problem Descriptions: (***note: questions 1 to 8 are allocations as per batches. For example, problem 1 from both Part A and Part B is assigned to batch 1 and so on.***)

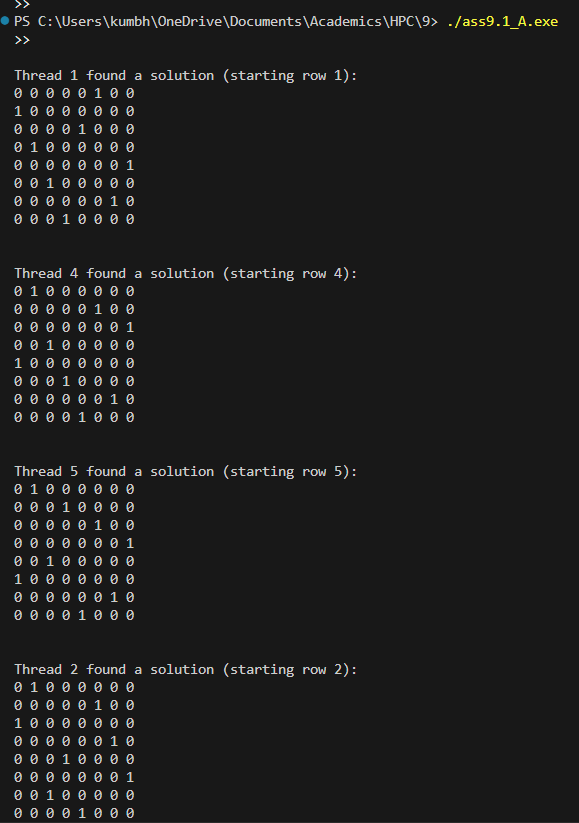
Part A — Exploratory Mini-Project:

1. N-Queens Problem – Parallelize backtracking; assign initial row placements to different threads.

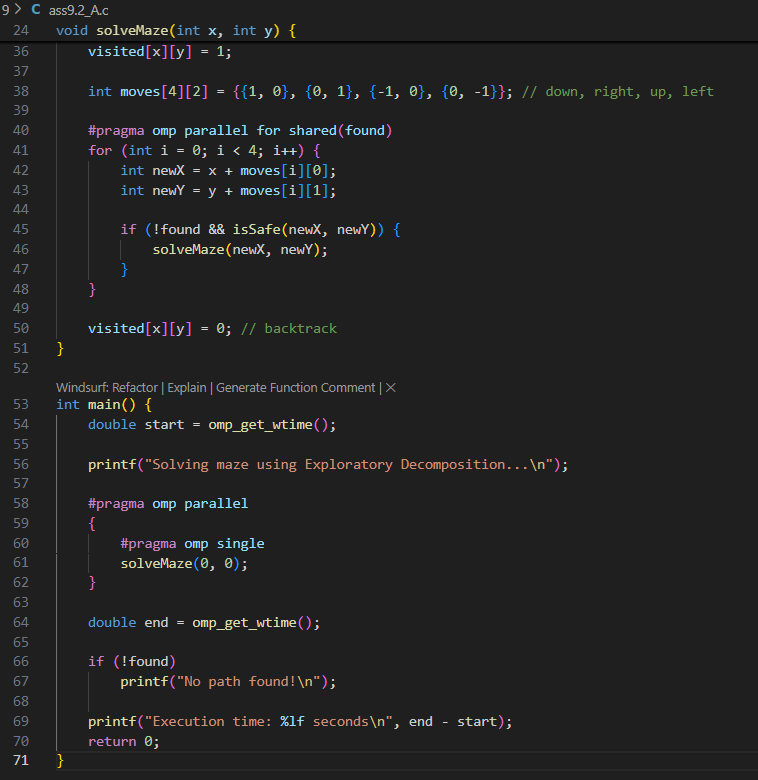
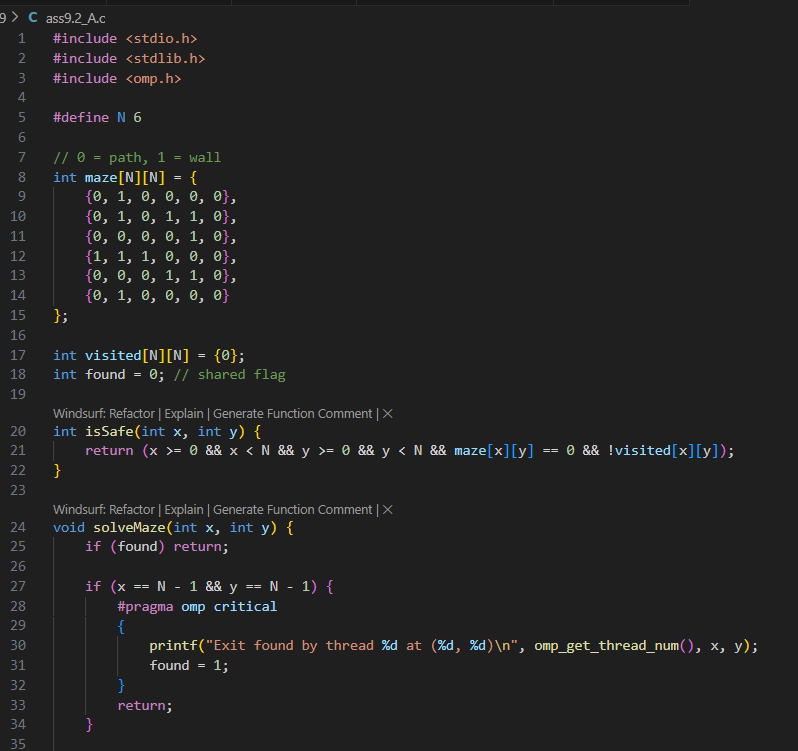
Part A.1 :

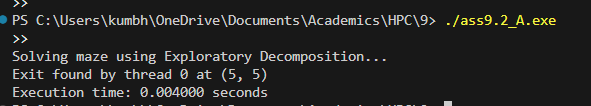


Results :

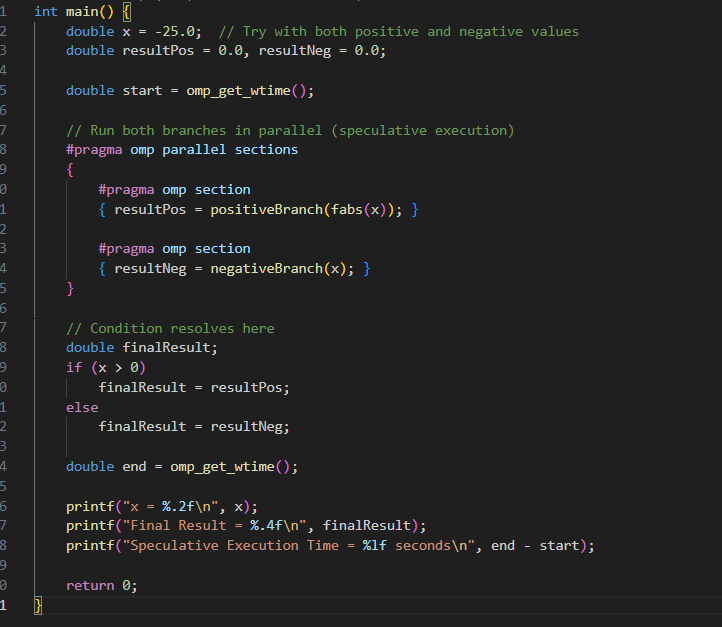
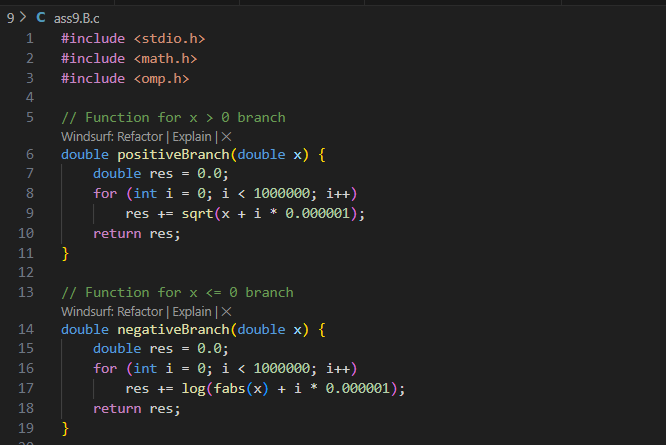


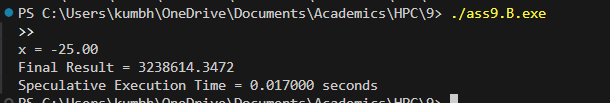
Part A.2 :





Part B — Speculative Mini-Project :

1. If–Else Branch Evaluation in Numerical Computation
   1. Suppose a function requires checking a condition (x > 0).
   2. Sequential: compute only one branch (sqrt(x) or log(|x|)).
   3. Speculative: compute both in parallel, then keep the correct one after condition resolves.
   4. 



**Report Submission:**

Prepare a short technical report (max 6 pages) including:

* Introduction to both techniques.
* Problem descriptions.
* Algorithm design with diagrams.
* Implementation details (code along with output snippets).
* Results (tables/graphs).
* Observations and conclusions.

Sample Results

| Problem | Sequential Time (ms) | Parallel Time (ms) | Speedup | Wasted Computation (%) |
| --- | --- | --- | --- | --- |
| N-Queens (Exploratory) | 850 | 300 | 2.83× | ~0% |
| Branch Execution (Speculative) | 950 | 520 | 1.82× | ~48% |